AUXILIARY VARIABLE MARKOV CHAIN MONTE CARLO METHODS

MATTHEW MCKENZIE GRAHAM

Doctor of Philosophy
University of Edinburgh
2017

Matthew Mckenzie Graham: *Auxiliary variable Markov chain Monte Carlo methods,* 2017.
Submitted for the degree of Doctor of Philosophy, University of Edinburgh.

SUPERVISOR:
Dr. Amos J. Storkey

ABSTRACT

Inference, the process of drawing conclusions from evidence, is at the heart of the scientific method. Probability theory offers a consistent framework for performing inference in the presence of uncertainty, posing the inference problem as the task of computing expectations of functions of interest with respect to a probability distribution.

In complex models with a large number of unknown variables to be inferred, these expecations can be intractable to compute exactly. This has motivated the development of a large class of approximate inference methods which tradeoff a lack of exactness for greater computational tractability. Monte Carlo methods are one class of such techniques, with the integrals or summations across the whole state space approximated by summations over a finite number of randomly sampled points. This maps the inference problem to that of drawing samples from, often complex, high-dimesional, probability distributions.

LAY SUMMARY

A lay summary is intended to facilitate knowledge exchange, public engagement and outreach. It should be in simple, non-technical terms that are easily understandable by a lay audience, who may be non-professional, non-scientific and outside the research area.

Abstracts, particularly in science, engineering, medicine and veterinary medicine, may be highly technical or contain scientific language that is not easily understandable to readers outside the research area. Therefore, the lay summary is intended as supplementary to the abstract.

ACKNOWLEDGEMENTS

Put acknowledgments here.

Lorem ipsum at nusquam appellantur his, ut eos erant homero concluda turque. Albucius appellantur deterruisset id eam, vivendum partiendo dissentiet ei ius. Vis melius facilisis ea, sea id convenire referrentur, takimata adolescens ex duo. Ei harum argumentum per. Eam vidit exerci appetere ad, ut vel zzril intellegam interpretaris.

Errem omnium ea per, pro con populo ornatus cu, ex qui dicant nemore melius. No pri diam iriure euismod. Graecis eleifend appellantur quo id. Id corpora inimicus nam, facer nonummy ne pro, kasd repudiandae ei mei. Mea menandri mediocrem dissentiet cu, ex nominati imperdiet nec, sea odio duis vocent ei. Tempor everti appareat cu ius, ridens audiam an qui, aliquid admodum conceptam ne qui. Vis ea melius nostrum, mel alienum euripidis eu.

DECLARATION

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Edinburgh, June 2017

Matthew Mckenzie Graham

CONTENTS

FRONT MATTER

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

MCMC	Markov chain Monte Carlo
KL	Kullback–Leibler
SDE	stochastic differential equation
HMC	Hamiltonian Monte Carlo
CAVI	coordinate ascent variational inference
SVI	stochastic variational inference
EP	expectation propagation
CPU	central processing unit
GPU	graphics processing unit
FLOPS	floating point operations per second
ELBO	evidence lower bound
PRNG	pseudo-random number generator
CDF	cumulative distribution function
ESS	effective sample size
iid	independently and identically distributed
wrt	with respect to
iff	if and only if

1 PROBABILISTIC MODELLING

The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. Therefore the true logic for this world is the calculus of probabilities
—James Clerk Maxwell

Inference is the process of drawing conclusions from evidence. Much of our lives are spent making inferences about the world given our observations of it; in particular inference is a central aspect of the scientific process. Although deductive logic offers a framework for inferring conclusions from absolute statements of truth, it does not apply to the more typical real-world setting where the information we receive is subject to uncertainty.

To make inferences under conditions of uncertainty, we must instead turn to probability theory. Probabilities offer a consistent framework for quantifying the uncertainty in our beliefs about the world and making inferences given these beliefs. The output of the inference process is itself probabilistic, reflecting that the conclusions we make given uncertain information will themselves be subject to uncertainty.

In this chapter we will first introduce the probability notation we will use in the rest of this work, and state some basic results which will be important in the later chapters. We will introduce graphical models as a compact way of visualising structure in probabilistic models. Finally we will give a concrete definition of the inference tasks that the methods presented in the rest of this thesis are aimed at computing (approximate) solutions to,and motivate why such approximate computational methods are needed.

1.1 PROBABILITY THEORY

A *probability space* is defined as a triplet $(S,\ \mathcal{E},\ \mathsf{P})$ where

- S is the *sample space*, the set of all possible outcomes,
- $\mathcal{E}$ is the *event space*, a σ-algebra on S, defining all possible events (measurable subsets of S),
- P is the *probability measure*, a finite measure satisfying $\mathsf{P}(S) = 1$, which specifies the probabilities of events in $\mathcal{E}$.

A σ-algebra, $\mathcal{E}$, on a set S is set of subsets of S with $S \in \mathcal{E}$, $\emptyset \in \mathcal{E}$ and which is closed under complement and countable unions and intersections.

Kolmogorov's axioms:

1. *Non-negativity:* $\mathsf{P}(E) \geq 0\ \forall E \in \mathcal{E}$,
2. *Normalisation:* $\mathsf{P}(S) = 1$,
3. *Countable additivity: for any countable set of disjoint events* $\{E_i\}_i : E_i \in \mathcal{F}\ \forall i$, $E_i \cap E_j = \emptyset\ \forall i \neq j$, $\mathsf{P}(\cup_i E_i) = \sum_i \mathsf{P}(E_i)$.

Given this definition of a probability space, Kolmogorov's axioms [59] can be used to derive a measure-theoretic formulation of probability theory. The probability of an event $E \in \mathcal{E}$ is defined as its measure $\mathsf{P}(E)$. Two events $A, B \in \mathcal{E}$ are *independent* if $\mathsf{P}(A \cap B) = \mathsf{P}(A)\mathsf{P}(B)$.

A measure-theoretic approach has the advantage of providing a unified treatment for describing probabilities on both finite and infinite sample spaces. Although alternative derivations of the laws of probability from different premises such as Cox's theorem [21, 22] have been proposed, modern extensions of this work result in a calculus of probabilities that is equivalent to Kolmogorov's [117], with the differences mainly being in the philosophical interpretations of probabilities.

1.1.1 Random variables

If $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ are two measurable spaces, a function $f : X \to Y$ is measurable if $f^{-1}(E) \in \mathcal{F}\ \forall E \in \mathcal{G}$.

When modelling real-world processes, rather than considering events as subsets of an abstract sample space, it is usually more helpful to consider *random variables* which represent quantities in the model of interest. A random variable $\mathsf{x} : S \to X$ is defined as a measurable function from the sample space to a measurable space $(X, \mathcal{F})$.

The Borel σ-algebra $\mathscr{B}(\mathbb{R})$ is the smallest σ-algebra on $\mathbb{R}$ which contains all open real intervals.

Often X is the reals, $\mathbb{R}$, and $\mathcal{F}$ is the Borel σ-algebra on the reals, $\mathscr{B}(\mathbb{R})$, in which case we will refer to a *real random variable*. It is also common to consider cases where X is a real vector space, $\mathbb{R}^D$, and $\mathcal{F} = \mathscr{B}(\mathbb{R}^D)$ - in this case we will term the resulting random variable a *random vector* and use the notation $\mathbf{x} : S \to X$. A final special case is when X is countable and $\mathcal{F}$ is the power set $\mathscr{P}(X)$ in which case we will refer to x as a *discrete random variable.*

If $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ are two measurable spaces, μ a measure on these spaces and $f : X \to Y$ a measurable function, the pushforward measure μ_f satisfies $\mu_f(A) = \mu \circ f^{-1}(A)$ $\forall A \in \mathcal{G}$.

Due to the definition of a random variable as a measurable function, we can define a pushforward measure on a random variable x

$$\mathsf{P}_{\mathsf{x}}(A) = \mathsf{P} \circ \mathsf{x}^{-1}(A) = \mathsf{P}(\{s \in S : \mathsf{x}(s) \in A\}) \quad \forall A \in \mathcal{F}. \tag{1.1}$$

The measure P_{x} specifies that the probability of the event that the random variable x takes a value in a measurable set $A \in \mathcal{F}$ is $\mathsf{P}_{\mathsf{x}}(A)$. We will sometimes describe P_{x} as the *probability distribution* of x.

1.1.2 Joint and conditional probability

Often we will jointly define multiple random variables on the same probability space. Let $(S, \mathcal{E}, \mathsf{P})$ be a probability space and $\mathsf{x} : S \to X$,

$\mathsf{y} : S \to Y$ be two random variables with corresponding σ-algebras $\mathcal{F}$ and $\mathcal{G}$. Then the *joint probability* of x and y is defined as

$$\mathsf{P}_{\mathsf{x},\mathsf{y}}(A,\,B) = \mathsf{P}\left(\mathsf{x}^{-1}(A) \cap \mathsf{y}^{-1}(B)\right) \quad \forall A \in \mathcal{F},\, B \in \mathcal{G}. \tag{1.2}$$

The joint probability is related to P_{x} and P_{y} by

$$\mathsf{P}_{\mathsf{x},\mathsf{y}}(A,\,Y) = \mathsf{P}_{\mathsf{x}}(A),\; \mathsf{P}_{\mathsf{x},\mathsf{y}}(X,\,B) = \mathsf{P}_{\mathsf{y}}(B) \quad \forall A \in \mathcal{F},\, B \in \mathcal{G}. \tag{1.3}$$

In this context P_{x} and P_{y} are referred to as *marginals* of the joint.

The two random variables are said to be independent if and only if

$$\mathsf{P}_{\mathsf{x},\,\mathsf{y}}(A, B) = \mathsf{P}_{\mathsf{x}}(A)\mathsf{P}_{\mathsf{y}}(B) \quad \forall A \in \mathcal{F},\, B \in \mathcal{G}. \tag{1.4}$$

Also useful is the definition of *conditional probability*

$$\mathsf{P}(A \,|\, B) = \frac{\mathsf{P}(A \cap B)}{\mathsf{P}(B)} \quad \forall A \in \mathcal{E},\, B \in \mathcal{E} : \mathsf{P}(B) \neq 0. \tag{1.5}$$

In Kolmogorov's probability theory, (1.5) is given as an additional definition distinct from the basic axioms. In alternatives such as the work of Cox [21, 22] and de Finetti [29], conditional probabilities are instead viewed as a primitive.

Correspondingly, the conditional probabilities of random variables $\mathsf{P}_{\mathsf{x}|\mathsf{y}}$ and $\mathsf{P}_{\mathsf{x}|\mathsf{y}}$ can likewise be defined as satisfying

$$\begin{aligned}\mathsf{P}_{\mathsf{x},\mathsf{y}}(A,\,B) = \mathsf{P}_{\mathsf{x}|\mathsf{y}}(A \,|\, B)\,\mathsf{P}_{\mathsf{y}}(B) = \mathsf{P}_{\mathsf{y}|\mathsf{x}}(B \,|\, A)\,\mathsf{P}_{\mathsf{x}}(A)\\ \forall A \in \mathcal{F},\, B \in \mathcal{G} : \mathsf{P}_{\mathsf{x},\mathsf{y}}(A,\,B) \neq 0,\end{aligned} \tag{1.6}$$

which is sometimes referred to as the product rule.

An implication of (1.6) is what is often termed *Bayes' theorem*

$$\mathsf{P}_{\mathsf{x}|\mathsf{y}}(A \,|\, B) = \frac{\mathsf{P}_{\mathsf{y}|\mathsf{x}}(B \,|\, A)\,\mathsf{P}_{\mathsf{x}}(A)}{\mathsf{P}_{\mathsf{y}}(B)} \quad \forall A \in \mathcal{F},\, B \in \mathcal{G} : \mathsf{P}_{\mathsf{y}}(B) \neq 0, \tag{1.7}$$

which will be of key importance in the later discussion of inference.

The definition in (1.2) of the joint probability of a pair of random variables can be extended to arbitarily large collections of random variables. Similarly conditional probabilities can be defined for collections of multiple jointly dependent random variables, with the product rule given in (1.6) generalising to a combinatorial number of possible factorisations of the joint probability. Graphical models offer a convenient way of representing the dependencies between large collections of random variables and any resulting factorisation structure in their joint probability, and will be discussed later in this chapter in section 1.2 .

1.1.3 Probability densities

So far we have ignored how the probability measure P is defined and by consequence the probability of a random variable. The Radon–Nikodym theorem guarantees that for a pair of σ–finite measures μ and ν on a measurable space $(X, \mathcal{F})$ where ν is absolutely continuous with respect to μ, then there is a unique (up to μ-null sets) measurable function $f : X \to [0, \infty)$ termed a *density* such that

A measure on X is σ-finite if X is a countable union of finite measure sets.

$$\nu(A) = \int_A f \,\mathrm{d}\mu \quad \forall A \in \mathcal{F}. \tag{1.8}$$

The density function f is also termed the *Radon-Nikodym derivative* of ν with respect to μ, denoted $\frac{\mathrm{d}\nu}{\mathrm{d}\mu}$. Density functions therefore represent a convenient way to define a probability measure with respect to an appropriate base measure. It can also be shown that if $f = \frac{\mathrm{d}\nu}{\mathrm{d}\mu}$ and g is a measurable function that

If μ and ν are measures on a measurable space $(X, \mathcal{F})$ then ν has absolute continuity wrt to μ if $\forall A \in \mathcal{F}$, $\mu(A) = 0 \Rightarrow \nu(A) = 0$.

$$\int_X g \,\mathrm{d}\nu = \int_X g\, f \,\mathrm{d}\mu, \tag{1.9}$$

which we will use later when discussing calculation of expectations.

For real random variables, an appropriate base measure is the *Lebesgue measure*, λ, on $\mathbb{R}$. The probability P_{x} of a real random variable x can then be defined via a *probability density* $\mathrm{p}_{\mathrm{x}} : \mathbb{R} \to [0, \infty)$ by

$$\mathsf{P}_{\mathrm{x}}(A) = \int_A \mathrm{p}_{\mathrm{x}} \,\mathrm{d}\lambda = \int_A \mathrm{p}_{\mathrm{x}}(x) \,\mathrm{d}x \qquad \forall A \in \mathscr{B}(\mathbb{R}). \tag{1.10}$$

Analagously for a random vector $\mathbf{x}$ with density $\mathrm{p}_{\mathbf{x}} : \mathbb{R}^D \to [0, \infty)$ with respect to the D-dimensional Lebesgue measure λ^D

$$\mathsf{P}_{\mathbf{x}}(A) = \int_A \mathrm{p}_{\mathbf{x}} \,\mathrm{d}\lambda^D = \int_A \mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \qquad \forall A \in \mathscr{B}(\mathbb{R}^D). \tag{1.11}$$

The notation in the second equalities in (1.10) and (1.11) uses a convention that will be used throughout this thesis that integrals without an explicit measure are with respect to the Lebesgue measure.

For discrete random variables, an appropriate base measure is instead the *counting measure*, $\#$. The probability of a discrete random variable

The counting measure $\#$ is defined as $\#(A) = |A|$ for all finite A and $\#(A) = +\infty$ otherwise.

is then defined via a probability density $\mathsf{p}_\mathsf{x} : X \to [0, 1]$ by

$$\mathsf{P}_\mathsf{x}(A) = \int_A \mathsf{p}_\mathsf{x} \, \mathrm{d}\# = \sum_{x \in A} \mathsf{p}_\mathsf{x}(x) \qquad \forall A \in \mathscr{P}(X). \tag{1.12}$$

The co-domain of a probability density p_x for a discrete random variable is restricted to $[0, 1]$ due to the non-negativity and normalisation requirements for the probability measure P_x, with $\sum_{x \in X} \mathsf{p}_\mathsf{x}(x) = 1$. Commonly for the case of a discrete random variable, the density p_x is instead referred to as a *probability mass function*, with density reserved for real random variables. We will however use *probability density* in both cases in keeping with the earlier definition of a density with respect to a base measure, this avoiding difficulties when definining joint probabilities on a mixture of real and discrete random variables.

The joint probability $\mathsf{P}_{\mathsf{x},\mathsf{y}}$ of a pair of random variables x and y with co-domains the measurable spaces $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ respectively, can be defined via a joint probability density $\mathsf{p}_{\mathsf{x},\mathsf{y}} : X \times Y \to [0, \infty)$ by

$$\mathsf{P}_{\mathsf{x},\mathsf{y}}(A, B) = \int_{A \times B} \mathsf{p}_{\mathsf{x},\mathsf{y}} \, \mathrm{d}(\mu_\mathsf{x} \times \mu_\mathsf{y}) \quad \forall A \in \mathcal{F},\ B \in \mathcal{G}, \tag{1.13}$$

where $\mu_\mathsf{x} \times \mu_\mathsf{y}$ represents the product measure of two appropriate base measures μ_x and μ_y, e.g. $\mu_\mathsf{x} = \lambda$ and $\mu_\mathsf{y} = \#$ if x is a real random variable and y is a discrete random variable.

If $(X_1, \mathcal{F}_1, \mu_1)$ and $(X_2, \mathcal{F}_2, \mu_2)$ are two measure spaces, the product measure $\mu_1 \times \mu_2$ on a measurable space $(X_1 \times X_2, \mathcal{F}_1 \otimes \mathcal{F}_2)$ is defined as satisfying $(\mu_1 \times \mu_2)(A_1 \times A_2) = \mu_1(A_1)\mu_2(A_2)$ $\forall A_1 \in \mathcal{F}_1,\ A_2 \in \mathcal{F}_2$.

When dealing with random variables, we will often only specify the co-domain of the random variable(s) and a (joint) probability density, with the base measure being implicitly defined as the Lebesgue measure for real random variables (or vectors), counting measure for discrete random variables and an appropriate product measure for a mix of random variables. Similarly we will usually neglect to explicitly define the probability space $(S, \mathcal{E}, \mathsf{P})$ which the random variable(s) map from. In this case we will typically use the loose notation $\mathsf{x} \in X$ to mean a random variable x with co-domain X.

This less explicit but more succinct probability notation in terms of random variables and densities is common in the machine learning and computational statitistics literature and will generally be preferred to improve readability. Tables 1.1, 1.2 and 1.3 give definitions of the densities and shorthand notation for some common parametric probability distributions that we will use in this thesis.

Name	Parameters	Shorthand	Density	Support
Bernoulli	$\pi \in [0,1]$	$\mathrm{Ber}(x \mid \pi)$	$\pi^x (1-\pi)^{(1-x)}$	$x \in \{0,1\}$
Categorical	$\boldsymbol{\pi} \in \mathbb{S}^K$	$\mathrm{Cat}(x \mid \boldsymbol{\pi})$	$\sum_{k=1}^{K} (\mathbb{1}_{\{k\}}(x) \pi_k)$	$x \in \{1 \ldots K\}$

Table 1.1: Definitions of densities of parametric distributions for discrete random variables that will be used in this thesis.

Name	Parameters	Shorthand	Density
Normal	$\mu \in \mathbb{R}$: mean $\sigma > 0$: standard deviation	$\mathcal{N}\left(x \mid \mu, \sigma^2\right)$	$\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
Multivariate normal	$\boldsymbol{\mu} \in \mathbb{R}^D$: mean vector $\Sigma \in \mathcal{S}_{++}^D$: covariance matrix	$\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma)$	$\frac{1}{\sqrt{(2\pi)^D \lvert\Sigma\rvert}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\mathsf{T} \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right)$
Student's t	$\nu > 0$: degrees of freedom $\mu \in \mathbb{R}$: location $\sigma > 0$: scale	$\mathrm{StT}(x \mid \nu, \mu, \sigma)$	$\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\pi\nu}\sigma} \left(1 + \frac{1}{\nu}\left(\frac{x-\mu}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}$
Logistic	$\mu \in \mathbb{R}$: location $\sigma > 0$: scale	$\mathrm{Logistic}(x \mid \mu, \sigma)$	$\frac{1}{4\sigma} \cosh\left(\frac{x-\mu}{2\sigma}\right)^{-2}$
Inverse cosh	$\mu \in \mathbb{R}$: location $\sigma > 0$: scale	$\mathrm{InvCosh}(x \mid \mu, \sigma)$	$\frac{1}{2\sigma} \cosh\left(\frac{\pi(x-\mu)}{2\sigma}\right)^{-1}$

Table 1.2: Definitions of densities of parametric distributions for unbounded real random variables that will be used in this thesis.

Name	Parameters	Shorthand	Density	Support
Log-normal	$\mu \in \mathbb{R}$: log mean $\sigma > 0$: log standard deviation	$\mathrm{LogNorm}(x \mid \mu, \sigma^2)$	$\frac{1}{x\sqrt{2\pi}\sigma} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right)$	$x > 0$
Multivariate log-normal	$\boldsymbol{\mu} \in \mathbb{R}^D$: log mean $\boldsymbol{\Sigma} \in \mathcal{S}_{++}^D$: log covariance	$\mathrm{LogNorm}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$	$\frac{\exp\left(-\frac{1}{2}(\log \boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\log \boldsymbol{x} - \boldsymbol{\mu})\right)}{\prod_{d=1}^D (x_d)\sqrt{(2\pi)^D \lvert\boldsymbol{\Sigma}\rvert}}$	$\boldsymbol{x} \in [0, \infty)^D$
Exponential	$\lambda > 0$: rate	$\mathrm{Exp}(x \mid \lambda)$	$\lambda \exp(-\lambda x)$	$x \geq 0$
Uniform	$a \in \mathbb{R}$: minimum $b \in \mathbb{R}$: maximum, $b > a$	$\mathcal{U}(x \mid a, b)$	$\frac{1}{b-a} \mathbb{1}_{[a,b]}(x)$	$a \leq x \leq b$
Half-Cauchy	$\gamma > 0$: scale	$\mathcal{C}_{\geq 0}(x \mid \gamma)$	$\frac{2}{\pi\gamma}\left(1 + \frac{x^2}{\gamma^2}\right)^{-1}$	$x \geq 0$
Gamma	$\alpha > 0$: shape $\beta > 0$: rate	$\mathrm{Gamma}(x \mid \alpha, \beta)$	$\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$	$x \geq 0$
Beta	$\alpha > 0$: shape $\beta > 0$: shape	$\mathrm{Beta}(x \mid \alpha, \beta)$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$	$0 \leq x \leq 1$
Dirichlet	$\boldsymbol{\alpha} \in (0, \infty)^K$: concentration	$\mathrm{Dir}(\boldsymbol{x} \mid \boldsymbol{\alpha})$	$\frac{\Gamma\left(\sum_{k=1}^K \alpha_k\right)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K x_i^{\alpha_k - 1}$	$\boldsymbol{x} \in \mathbb{S}^K$
Lomax	$\alpha > 0$: shape $\beta > 0$: scale	$\mathrm{Lomax}(x \mid \alpha, \beta)$	$\frac{\alpha\beta^\alpha}{(\beta+x)^{\alpha+1}}$	$x \geq 0$

Table 1.3: Definitions of densities of parameteric distributions for bounded real random variables that will be used in this thesis.

1.1.4 Transforms of random variables

It is common to define a random variable via a transform of another. Let x be a random variable with co-domain the measurable space $(X, \mathcal{F})$. Further let $(Y, \mathcal{G})$ be a second measurable space and $\phi : X \to Y$ a measurable function between the two spaces. If we define $\mathsf{y} = \phi \circ \mathsf{x}$ then analagously to our original definition of $\mathsf{P_x}$ as the pushforward measure of P under the measurable function defining x, we can define $\mathsf{P_y}$ in terms of $\mathsf{P_x}$ as

$$\mathsf{P_y}(A) = \mathsf{P_x} \circ \phi^{-1}(A) = \mathsf{P_x}(\{x \in X : \phi(x) \in A\}) \quad \forall A \in \mathcal{G}, \tag{1.14}$$

i.e. the probability of the event $\mathsf{y} \in A$ is equal to the probability of x being in the pre-image under ϕ of A. To calculate probabilities of transformed random variables therefore we will therefore need to be able to find the pre-images of values of the transformed variable.

If the probability $\mathsf{P_x}$ is defined by a probability density $\mathsf{p_x}$ with respect to a measure μ_x, we can also in some cases find a density $\mathsf{p_y}$ on the transformed variable $\mathsf{y} = \phi(\mathsf{x})$ with respect to a (potentially different) measure μ_y which can be used to calculate the probability $\mathsf{P_y}$,

$$\mathsf{P_y}(A) = \int_{\phi^{-1}(A)} \mathsf{p_x}\,\mathrm{d}\mu_\mathsf{x} = \int_A \mathsf{p_y}\,\mathrm{d}\mu_\mathsf{y} \quad \forall A \in \mathcal{G}. \tag{1.15}$$

For random variables with countable co-domains where the integral in (1.15) corresponds to a sum, a $\mathsf{p_y}$ satisfying (1.15) is simple to identify. If x is a discrete random variable with probability density $\mathsf{p_x}$ with respect to the counting measure, then $\mathsf{y} = \phi(\mathsf{x})$ will necessarily also be a discrete random variable. Applying (1.15) for $\mathsf{p_x} = \frac{\mathrm{d}\mathsf{P_x}}{\mathrm{d}\#}$ we have that

$$\begin{aligned}\int_{\phi^{-1}(A)} \mathsf{p_x}(x)\,\mathrm{d}\#(x) &= \sum_{x \in \phi^{-1}(A)} \mathsf{p_x}(x) = \sum_{y \in A} \sum_{x \in \phi^{-1}(y)} \mathsf{p_x}(x) \\ &= \int_A \sum_{x \in \phi^{-1}(y)} \mathsf{p_x}(x)\,\mathrm{d}\#(y) \quad \forall A \in \mathcal{G}. \end{aligned} \tag{1.16}$$

We can therefore define $\mathsf{p_y} = \frac{\mathrm{d}\mathsf{P_y}}{\mathrm{d}\#}$ in terms of $\mathsf{p_x}$ as

$$\mathsf{p_y}(y) = \sum_{x \in \phi^{-1}(y)} \mathsf{p_x}(x) \quad \forall y \in Y. \tag{1.17}$$

In the special case that ϕ is bijective we have that

$$\mathrm{p}_{\mathrm{y}}(y) = \mathrm{p}_{\mathrm{x}} \circ \phi^{-1}(y) \quad \forall y \in Y. \tag{1.18}$$

For transformations of real random variables and vectors, the situation is more complicated as we need to account for any local contraction or expansion of space by the map ϕ. Let $X = \mathbb{R}^M$ and $Y = \mathbb{R}^N$ with $N \leq M$, $N, M \in \mathbb{N}$. We will need a result from geometric measure theory, the *co-area formula* [28]. Let g be an L^1 integrable function and $\boldsymbol{\phi} : X \to Y$ a Lipschitz map. Then the co-area formula states that

$$\int_X g(\boldsymbol{x})\, J_{\boldsymbol{\phi}}(\boldsymbol{x})\, \mathrm{d}\lambda^M(\boldsymbol{x}) = \int_Y \int_{\boldsymbol{\phi}^{-1}(\boldsymbol{y})} g(\boldsymbol{x})\, \mathrm{d}\mathcal{H}^{M-N}(\boldsymbol{x})\, \mathrm{d}\lambda^N(\boldsymbol{y}) \tag{1.19}$$

where $\mathcal{H}^D$ is the D-dimensional *Hausdorff measure* and $J_{\boldsymbol{\phi}} : X \to [0, \infty)$ is the *Jacobian determinant* defined as

$$J_{\boldsymbol{\phi}}(\boldsymbol{x}) = \left| \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}}^{\mathsf{T}} \right|^{\frac{1}{2}} \quad \forall \boldsymbol{x} \in X. \tag{1.20}$$

The D-dimensional Hausdorff measure $\mathcal{H}^D$ on $\mathbb{R}^N$ for $D \in \mathbb{N}, 0 < D < N$ formalises a measure of the 'volume' of D-dimensional submanifolds of $\mathbb{R}^N$ - e.g. for $D = 1$ it corresponds to the length of a curve in $\mathbb{R}^N$. Additionally $\mathcal{H}^N = \lambda^N$ and $\mathcal{H}^0 = \#$.

Now let $\mathbf{x}$ be a random vector with co-domain the measurable space $(X, \mathscr{B}(\mathbb{R}^M))$ and define $\mathbf{y} = \boldsymbol{\phi} \circ \mathbf{x}$ as a random vector with co-domain the measurable space $(Y, \mathscr{B}(\mathbb{R}^N))$ with $\boldsymbol{\phi} : X \to Y$ a Lipschitz map as above. Let $Z = \left\{\boldsymbol{x} \in X : J_{\boldsymbol{\phi}}(\boldsymbol{x}) = 0\right\}$ and require that $\mathrm{P}_{\mathrm{x}}(Z) = 0$. Then for $A \in \mathscr{B}(\mathbb{R}^N)$ define an L^1 integrable function g as

$$g(\boldsymbol{x}) = \begin{cases} \mathbb{1}_A \circ \boldsymbol{\phi}(\boldsymbol{x})\, \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})\, J_{\boldsymbol{\phi}}(\boldsymbol{x})^{-1} & \forall \boldsymbol{x} \in X \setminus Z \\ 0 & \forall \boldsymbol{x} \in Z \end{cases}. \tag{1.21}$$

Integrating $g(\boldsymbol{x})\, J_{\boldsymbol{\phi}}(\boldsymbol{x})$ over $\boldsymbol{x} \in X$ we have that

$$\int_X g(\boldsymbol{x})\, J_{\boldsymbol{\phi}}(\boldsymbol{x})\, \mathrm{d}\lambda^M(\boldsymbol{x}) = \int_{X \setminus Z} \mathbb{1}_A \circ \boldsymbol{\phi}(\boldsymbol{x})\, \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})\, \mathrm{d}\lambda^M(\boldsymbol{x}) \tag{1.22}$$

$$= \int_X \mathbb{1}_A \circ \boldsymbol{\phi}(\boldsymbol{x})\, \mathrm{d}\mathrm{P}_{\mathrm{x}}(\boldsymbol{x}) \tag{1.23}$$

$$= \int_{\phi^{-1}(A)} \mathrm{d}\mathrm{P}_{\mathrm{x}}(\boldsymbol{x}) = \mathrm{P}_{\mathrm{y}}(A). \tag{1.24}$$

The equality between first and second lines comes from the requirement $\mathrm{P}_{\mathrm{x}}(Z) = 0$, with the Lebesgue integrals of a function over two sets which differ by only a zero-measure set equal. Now applying the co-area formula (1.19) to the left-hand side gives

$$\int_Y \int_{\boldsymbol{\phi}^{-1}(\boldsymbol{y})} g(\boldsymbol{x})\,\mathrm{d}\mathcal{H}^{M-N}(\boldsymbol{x})\,\mathrm{d}\lambda^N(\boldsymbol{y}) = \mathsf{P}_{\mathsf{y}}(A). \tag{1.25}$$

Therefore we can define a density $\mathsf{p}_{\mathsf{y}} = \frac{\mathrm{d}\mathsf{P}_{\mathsf{y}}}{\mathrm{d}\lambda^N}$ satisfying (1.15) as

$$\mathsf{p}_{\mathsf{y}}(\boldsymbol{y}) = \int_{\boldsymbol{\phi}^{-1}(\boldsymbol{y})} \mathsf{p}_{\mathbf{x}}(\boldsymbol{x})\, J_{\boldsymbol{\phi}}(\boldsymbol{x})^{-1}\,\mathrm{d}\mathcal{H}^{M-N}(\boldsymbol{x}) \quad \forall \boldsymbol{y} \notin \boldsymbol{\phi}(Z). \tag{1.26}$$

For the special case of a dimension-preserving map $\boldsymbol{\phi}$ with $N = M$ the integral in (1.26) is with respect to $\mathcal{H}^0$ which is equivalent to the counting measure #. In this case $J_{\boldsymbol{\phi}}(\boldsymbol{x}) = \left|\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}}\right|$ and we therefore get

$$\mathsf{p}_{\mathsf{y}}(\boldsymbol{y}) = \sum_{\boldsymbol{x} \in \boldsymbol{\phi}^{-1}(\boldsymbol{y})} \mathsf{p}_{\mathbf{x}}(\boldsymbol{x}) \left|\frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{x}}\right|^{-1} \quad \forall \boldsymbol{y} \notin \boldsymbol{\phi}(Z). \tag{1.27}$$

Under the further restriction that $\boldsymbol{\phi}$ is bi-Lipschitz, i.e. it is bijective and Lipschitz in both directions, we recover the more commonly presented multidimensional change of variables formula

$$\mathsf{p}_{\mathsf{y}}(\boldsymbol{y}) = \mathsf{p}_{\mathbf{x}} \circ \boldsymbol{\phi}^{-1}(\boldsymbol{y}) \left|\frac{\partial \boldsymbol{\phi}^{-1}}{\partial \boldsymbol{y}}\right| \quad \forall \boldsymbol{y} \in Y. \tag{1.28}$$

In both of the cases considered, we have seen that if the function ϕ the random variable x is mapped through is bijective, the resulting expression for the density on the mapped random variable y is simpler in the sense that the pre-image $\phi^{-1}(y)$ of a point $y \in Y$ is itself a point and so we do not need to integrate or sum over points in the pre-image which will often be difficult to do analytically.

Bijectivity is a very limiting condition however, with many models involving non-bijective transformations of random variables. Later in this thesis we will see that methods used for defining the more general forms for calculating the density of a transformed variable are key to proposed methods for performing inference in generative models defined by complex, non-dimension preserving and non-bijective transformations of random variables.

1.1.5 Expectations

A key operation when working with probabilistic models is computing expectations. Let $(S, \mathcal{E}, \mathrm{P})$ be a probability space, and $\mathrm{x} : S \to X$ a random variable on this space. The *expected value of* x is defined as

$$\mathbb{E}[\mathrm{x}] = \int_S \mathrm{x}(s)\,\mathrm{dP}(s). \tag{1.29}$$

Often it will be more convenient to express expectations in terms of the probability P_x instead. If $f : S \to X$ is a measurable function and μ a measure on S then the integral with respect to the pushforward measure μ_f of an integrable function g satisfies

$$\int_X g(x)\,\mathrm{d}\mu_f(x) = \int_S g \circ f(s)\,\mathrm{d}\mu(s). \tag{1.30}$$

If we take g as the identity map we therefore have that

$$\mathbb{E}[\mathrm{x}] = \int_X x\,\mathrm{dP}_\mathrm{x}(x). \tag{1.31}$$

If P_x is given by a density $\mathrm{p}_\mathrm{x} = \frac{\mathrm{dP}_\mathrm{x}}{\mathrm{d}\mu}$ then using (1.9) we also have

$$\mathbb{E}[\mathrm{x}] = \int_X x\,\mathrm{p}_\mathrm{x}(x)\,\mathrm{d}\mu(x), \tag{1.32}$$

which is often the form used for computation.

A further useful implication of (1.30) is what is sometimes termed the *Law of the unconscious statistician*. Let $\mathrm{x} : S \to X$ be a random variable, $\phi : X \to Y$ a measurable function and define $\mathrm{y} = \phi \circ \mathrm{x}$. Then

$$\mathbb{E}[\mathrm{y}] = \int_S \mathrm{y}(s)\,\mathrm{dP}(s) = \int_S \phi \circ \mathrm{x}(s)\,\mathrm{dP}(s) = \int_X \phi(x)\,\mathrm{dP}_\mathrm{x}(x), \tag{1.33}$$

i.e. it can be calculated by integrating ϕ with respect to P_x. This means we can calculate expectations of a transformed random variable $\mathrm{y} = \phi(\mathrm{x})$ without needing to use the change of variables formulae from Section 1.1.4 to explicitly calculate the probability P_y (or density p_y) and with a relatively weak condition of measurability on ϕ.

Related to the expected value are the *variance* and *covariance*, which for scalar random variables x and y are respectively defined as

$$\mathbb{V}[\mathrm{x}] = \mathbb{E}\left[(\mathrm{x} - \mathbb{E}[\mathrm{x}])^2\right] \text{ and } \mathbb{C}[\mathrm{x},\mathrm{y}] = \mathbb{E}[(\mathrm{x} - \mathbb{E}[\mathrm{x}])(\mathrm{y} - \mathbb{E}[\mathrm{x}])]. \tag{1.34}$$

1.1.6 Conditional expectations and densities

A related concept, and one which will be key in our discussion of inference, is conditional expectation. Let $(S, \mathcal{E}, \mathrm{P})$ be a probability space, $(X, \mathcal{F})$ and $(Y, \mathcal{G})$ two measurable spaces and $\mathrm{x} : S \to X$ and $\mathrm{y} : S \to Y$ two random variables. Then the *conditional expectation of* x *given* y, is defined as a measurable function $\mathbb{E}[\mathrm{x} \,|\, \mathrm{y}] : Y \to X$ satisfying

$$\int_{\mathrm{y}^{-1}(A)} \mathrm{x}(s)\,\mathrm{dP}(s) = \int_A \mathbb{E}[\mathrm{x} \,|\, \mathrm{y}](y)\,\mathrm{dP}_{\mathrm{y}}(y) \quad \forall A \in \mathcal{G}. \tag{1.35}$$

$\mathbb{E}[\mathrm{x} \,|\, \mathrm{y}]$ is guaranteed to be uniquely defined almost everywhere in Y by (1.35), i.e. up to P_{y}-null sets. As a particular case where $A = Y$ we recover what is sometimes termed the *Law of total expectation*

$$\int_S \mathrm{x}\,\mathrm{dP} = \int_S \mathbb{E}[\mathrm{x} \,|\, \mathrm{y}] \circ \mathrm{y}\,\mathrm{dP} \implies \mathbb{E}[\mathrm{x}] = \mathbb{E}[\mathbb{E}[\mathrm{x} \,|\, \mathrm{y}] \circ \mathrm{y}]. \tag{1.36}$$

We will also use an alternative notation for the conditional expectation evaluated at point $\mathbb{E}[\mathrm{x} \,|\, \mathrm{y} = y] \equiv \mathbb{E}[\mathrm{x} \,|\, \mathrm{y}](y)$ but use the latter in this section to stress its definition as a measurable function.

Conditional expectation can be used to define the *regular conditional probability distribution* of a random variable conditioned on another random variable

$$\mathrm{P}_{\mathrm{x}}(B \,|\, \mathrm{y}) = \mathbb{E}[\mathbb{1}_A \circ \mathrm{x} \,|\, \mathrm{y}] \quad \forall B \in \mathcal{F}. \tag{1.37}$$

Likewise we can use conditional expectation to motivate a definition of conditional density. Assume a joint density $\mathrm{p}_{\mathrm{x,y}} = \frac{\mathrm{dP}_{\mathrm{x,y}}}{\mathrm{d}(\mu_{\mathrm{x}} \times \mu_{\mathrm{y}})}$ exists and has marginal density $\mathrm{p}_{\mathrm{y}} = \frac{\mathrm{dP}_{\mathrm{y}}}{\mathrm{d}\mu_{\mathrm{y}}}$. Then for all $A \in \mathcal{G}$

$$\int_{\mathrm{y}^{-1}(A)} \mathrm{x}(s)\,\mathrm{dP}(s) = \int_S \mathrm{x}(s)\,\mathbb{1}_A \circ \mathrm{y}(s)\,\mathrm{dP}(s) \tag{1.38}$$

$$= \int_{X \times Y} x\,\mathbb{1}_A(y)\,\mathrm{dP}_{\mathrm{x,y}}(x, y) \tag{1.39}$$

$$= \int_A \int_X x\,\mathrm{p}_{\mathrm{x,y}}(x, y)\,\mathrm{d}\mu_{\mathrm{x}}(x)\,\mathrm{d}\mu_{\mathrm{y}}(y). \tag{1.40}$$

Define $g : Y \to X$ as

$$g(y) = \begin{cases} \int_X x\,\frac{\mathrm{p}_{\mathrm{x,y}}(x,y)}{\mathrm{p}_{\mathrm{y}}(y)}\,\mathrm{d}\mu_{\mathrm{x}}(x) & \forall y \in Y : \mathrm{p}_{\mathrm{y}}(y) > 0 \\ 0 & \forall y \in Y : \mathrm{p}_{\mathrm{y}}(y) = 0. \end{cases} \tag{1.41}$$

Then from (1.40) we have that for all $A \in \mathcal{G}$

$$\int_{\mathsf{y}^{-1}(A)} \mathsf{x}(s)\,\mathrm{dP}(s) = \int_A g(y)\,\mathrm{p_y}(y)\,\mathrm{d}\mu_{\mathsf{y}}(y) = \int_A g(y)\,\mathrm{dP_y}(y). \tag{1.42}$$

The definition of g in (1.41) therefore satisfies the definition of conditional expectation in (1.35) and is uniquely defined up to a $\mathrm{P_y}$-null set. Therefore if $\mathrm{p_{x,y}}$ and $\mathrm{p_y}$ can be defined we have that

$$\mathbb{E}[\mathsf{x}\,|\,\mathsf{y}](y) = \int_X x\,\mathrm{p_{x|y}}(x\,|\,y)\,\mathrm{d}\mu_{\mathsf{x}}(x) \quad \forall y \in Y : \mathrm{p_y}(y) > 0 \tag{1.43}$$

where the *conditional density of* x *given* y, $\mathrm{p_{x|y}}$, is defined as

$$\mathrm{p_{x|y}}(x\,|\,y) = \frac{\mathrm{p_{x,y}}(x,y)}{\mathrm{p_y}(y)} \quad \forall x \in X,\ y \in Y : \mathrm{p_y}(y) > 0 \tag{1.44}$$

which can be seen to be analagous to the definition of conditional probability in (1.5). Note the definition of conditional expectation in (1.35) was not dependent on a joint density $\mathrm{p_{x,y}}$ being defined.

1.2 GRAPHICAL MODELS

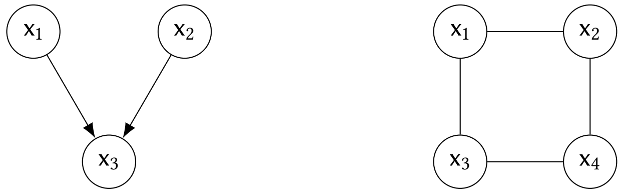
Graphical models = statistics × graph theory × computer science
—Zoubin Ghahramani

When working with probabilistic models involving large numbers of random variables, it will often be the case that not all the variables are jointly dependent on each other but that instead there are more local conditional relationships between them. Graphical models, which use graphs to describe the dependencies between random variables, are a useful framework for visualising the structure in complex probabilistic models and for giving a graph-theoretic basis for establishing the dependence between sets of random variables.

Central to all graphical models is the concept of conditional independence. Let $(S,\,\mathcal{E},\,\mathrm{P})$ be a probability space and $\mathsf{x} : S \to X$, $\mathsf{y} : S \to Y$ and $\mathsf{z} : S \to Z$ be three random variables with corresponding σ-algebras, $\mathcal{F}_{\mathsf{x}}$, $\mathcal{F}_{\mathsf{y}}$ and $\mathcal{F}_{\mathsf{z}}$ respectively. Following from our earlier definition of (unconditional) independence of random variables in (1.4), we say that x *and* y *are conditionally independent given* z, denoted $\mathsf{x} \perp \mathsf{y} \,|\, \mathsf{z}$, if

$$\mathrm{P_{x,y}}(A,B\,|\,\mathsf{z}) = \mathrm{P_x}(A\,|\,\mathsf{z})\,\mathrm{P_y}(B\,|\,\mathsf{z}) \ \ \forall A \in \mathcal{F}_{\mathsf{x}},\ B \in \mathcal{F}_{\mathsf{y}}, \tag{1.45}$$

holds almost everywhere with respect to $\mathrm{P_z}$.



(a) Directed graphical model. (b) Undirected graphical model.

Figure 1.1: Examples of directed and undirected graphical models. Circular nodes represent random variables in the model, with edges between them indicating dependencies between variables.

If a joint density on the random variables exists, a sufficient condition for $\mathsf{x} \perp \mathsf{y} \mid \mathsf{z}$ is that the conditional density $\mathsf{p}_{\mathsf{x},\mathsf{y}|\mathsf{z}}$ factorises as

$$\mathsf{p}_{\mathsf{x},\mathsf{y}|\mathsf{z}}(x, y \mid z) = \mathsf{p}_{\mathsf{x}|\mathsf{z}}(x \mid z)\mathsf{p}_{\mathsf{y}|\mathsf{z}}(y \mid z) \quad \forall x \in X,\ y \in Y,\ z \in Z. \tag{1.46}$$

This definition can be naturally extended to conditional independence when conditioning on more than one random variable, for example

$$\mathsf{v} \perp \mathsf{x} \mid \mathsf{y}, \mathsf{z} \implies \mathsf{p}_{\mathsf{v},\mathsf{x}|\mathsf{y},\mathsf{z}}(v, x \mid y, z) = \mathsf{p}_{\mathsf{v}|\mathsf{y},\mathsf{z}}(v \mid y, z)\mathsf{p}_{\mathsf{x}|\mathsf{y},\mathsf{z}}(x \mid y, z) \tag{1.47}$$

1.2.1 Directed and undirected graphical models

Several different graphical frameworks have been proposed for representing conditional independency relationships (and other information) in probabilistic models.

Directed graphical models [87], also known as *Bayesian networks*, represent probabilistic models as *directed acyclic graphs* (i.e. a directed graph in which there are no directed cycles), with the nodes in the graph representing random variables in the model and the edges of the graph defining a factorisation of the joint density over these variables into a product of conditional and marginal densities. In particular a conditional density factor is included for each node with parents (on the node random variable value given the parent variable values) and a marginal density factor for each root node without any parents.

An example directed graphical model for three random variables, x_1, x_2 and x_3, is shown in Figure 1.1a. The graph implies that the joint density can be factorised as

$$\mathsf{p}_{\mathsf{x}_1,\mathsf{x}_2,\mathsf{x}_3}(x_1, x_2, x_3) = \mathsf{p}_{\mathsf{x}_3|\mathsf{x}_1,\mathsf{x}_2}(x_3 \mid x_1, x_2)\, \mathsf{p}_{\mathsf{x}_1}(x_1)\, \mathsf{p}_{\mathsf{x}_2}(x_2). \tag{1.48}$$

Note that this factorisation would not be valid for all joint densities on the three variables; in particular we have that x_1 and x_2 are (unconditionally) independent and so that the joint density $\mathrm{p}_{\mathrm{x}_1,\mathrm{x}_2}$ can be written as the product of the two marginals $\mathrm{p}_{\mathrm{x}_1}$ and $\mathrm{p}_{\mathrm{x}_2}$.

Directed graphical models are a natural way of specifying *generative models* - i.e. probabilistic models which can be used to generate simulated observable quantities. Typically the factorisation specified by a directed graphical model gives a straightforward method to generate values from the joint density via *ancestral sampling*.

Ancestral sampling in a directed graphical model corresponds to first sampling values from all the root nodes from their marginal densities, then iteratively sampling from the conditional densities on each node for which all the parents nodes already have sampled values to condition on.

An alternative formalism for graphically representing probabilistic models is that of *undirected graphical models* [55], which are also known as *Markov random fields*. As with directed graphical models, each node in the graph represents a random variable, but here the edges connecting nodes are undirected. Rather than describing a factorisation of a joint density into conditional and marginal densities, an undirected graphical model indicates the factorisation of a joint density into a product of clique potentials on each of the maximal cliques in the graph. A *clique* is a fully connected component of the graph - i.e. a subset of nodes in the graph such that all pairs of nodes in the subset are connected by an edge. A *maximal clique* is a clique which is not a strict subset of any other clique. A *clique potential* is a non-negative function of the values of the random variables in the clique; it does necessarily correspond to any conditional or marginal probabilty density.

An example undirected graphical model on four random variables, x_1, x_2, x_3 and x_4, is shown in Figure 1.1b. Here the (maximal) cliques correspond to all the connected pairs of nodes. If $\psi_{a,b}$ denotes the clique potential on the pair (a, b) then the graphical model implies the joint density can be factorised as

$$\frac{1}{Z}\psi_{\mathrm{x}_1,\mathrm{x}_2}(x_1,x_2)\psi_{\mathrm{x}_1,\mathrm{x}_3}(x_1,x_3)\psi_{\mathrm{x}_2,\mathrm{x}_4}(x_2,x_4)\psi_{\mathrm{x}_3,\mathrm{x}_4}(x_3,x_4), \tag{1.49}$$

with Z a normalising constant such that the density integrates to 1 and so defines a valid probability measure.

Undirected graphical models are a natural representation for models of systems of mutually interacting components. For example they are commonly used in models of images to represent dependencies between pixel values and models of magnetic interactions in particle lattices.

Unlike directed models, generating joint configurations of the random variables in an undirected graphical model from the implied joint distribution is typically a non-trivial task, with no general equivalent to ancestral sampling. Further the joint density can typically only be evaluated up to an unknown normalising constant, with the integral needed to evaluate this constant often intractable for models involving a large number of variables or complex potentials. As we will see, these properties mean that inference in distributions defined by undirected graphical models is often particularly challenging.

D-separation: $\mathrm{x} \perp \mathrm{y} \,|\, C \iff$ *all paths in the graph between* x *and* y *are blocked. A path is blocked if at least one of the following holds: 1. The path includes a* →○→ *node or a* ←○→ *node in C. 2. The path includes a* →○← *node and neither the node or its descendants are in C.*

U-separation: x *and* y *in the model and a conditioning set of random variables* C, $\mathrm{x} \perp \mathrm{y} \,|\, C \iff$ *at least one random variable node on every path between* x *and* y *is in C.*

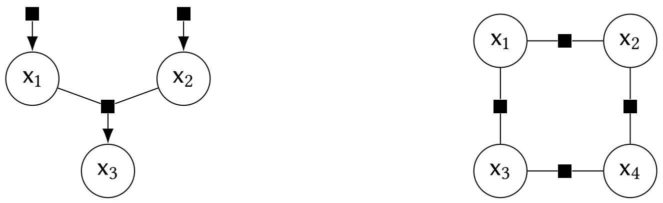
As suggested at the start of this section, both directed and undirected graphical models encode conditional independence properties of probabilistic models. In particular the rules of *D-separation* for directed graphical models and *U-separation* for undirected model give graph-based algorithmic descriptions of how to determine whether a pair of random variables are conditionally independent for a given conditioning set of random variables.

For example the directed graphical model in Figure 1.1a encodes the (un)conditional independence property $\mathrm{x}_1 \perp \mathrm{x}_2 \,|\, \emptyset = \mathrm{x}_1 \perp \mathrm{x}_2$ i.e. that x_1 and x_2 are independent if the value of x_3 is *not* conditioned on. The undirected graphical model in Figure 1.1b encodes the conditional independence properties $\mathrm{x}_1 \perp \mathrm{x}_4 \,|\, \mathrm{x}_2, \mathrm{x}_3$ and $\mathrm{x}_2 \perp \mathrm{x}_3 \,|\, \mathrm{x}_1, \mathrm{x}_4$.

Although there are methods to convert a directed graphical model to an undirected one and vice versa, in general these transformations are lossy - not all of the conditional independence relationships encoded in the original graph will necessarily be maintained in the transformed graph. For example there is no undirected graphical model which will represent the exact set of conditional independence properties represented by the directed graphical model in Figure 1.1a. Likewise there is no directed graphical model which will represent the exact set of conditional independence properties represented by the undirected graphical model in Figure 1.1b. Further there are distributions with dependency structures and factorisations which cannot be uniquely represented by either directed or undirected graphical models [30].

1.2.2 Factor graphs

An alternative graphical model formalism which overcomes some of the limitations of directed and undirected graphical models is that of



(a) A factor graph equivalent of the directed model in Figure 1.1a.

(b) A factor graph equivalent of the undirected model in Figure 1.1b.

Figure 1.2: Examples of factor graphs corresponding to the directed and undirected graphical models in Figure 1.1. Square black nodes correspond to individual factors depending on the connected variables (represented by circular nodes) in the joint density.

factor graphs [30, 31]. In factor graphs, in addition to nodes representing random variables, represented as in directed and undirected graphical models by circular nodes, a second class of nodes, denoted by filled squares (■), are introduced which represent individual factors in the joint density across the random variables represented in the model.

Factors may be either directed or undirected. Undirected factors, denoted by factor nodes in which all edges connecting to variable nodes are undirected, correspond to a factor in the joint density which depends on all of the variables with nodes connected to the factor, but without any requirement that the factor corresponds to a conditional or marginal probability density. Directed factors, denoted by factor nodes in which at least one edge from the factor node to a variable node is directed, correspond to a conditional density on the variables pointed to by directed edges given the values of the variables connected to the factor node by undirected edges (if there are no such variables then the factor instead corresponds to a marginal density).

Edges between nodes in a factor graph are always between nodes of disparate types i.e. between factor and variable nodes, but never between factor and factor or variable and variable nodes. As with directed graphical models, factor graphs with directed factors must not contain any directed cycles (i.e. a connected loop of edges in which one of every pair of edges connected to a factor on the loop is directed and all of the directed edges point in the same sense around the loop).

In the original extension of undirected factor graphs [31] to include directivity [30], it was proposed to allow multiple directed factors to connect via directed edges to the same variable node, representing multiple

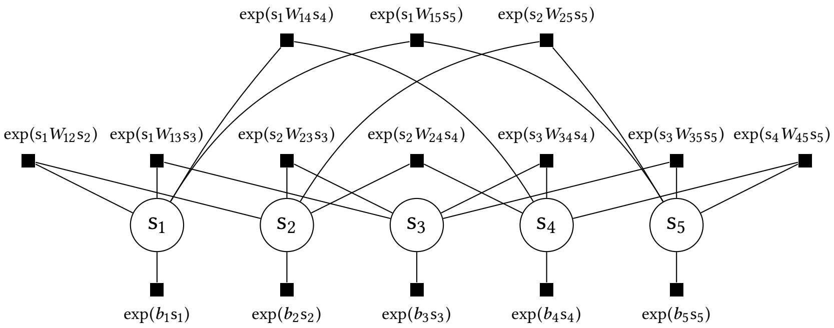


Figure 1.3: Five unit Boltzmann machine factor graph showing explicit factorisation of distribution into pairwise and single variable potentials.

factors in a conditional density on that varaiable. This generalisation introduces extra normalisation requirements and looses the interpretation of a directed factor as directly representing a conditional density, and so we will here only use directed factor graphs in which there is at most one directed edge connecting from a factor to a node.

Whether two variables are conditionally independent given a set of other variables can be checked from a factor graph by checking if all paths (i.e. connected series of edges and nodes) between the two corresponding variables nodes in the factor graph are *blocked*. A path is blocked if at least one of the following conditions is satisfied [30]

1. One of the variable nodes in the path is in the conditioning set.
2. One of the directed factor nodes in the path has two connected undirected edges in the path and there is no second directed path from the node to a variable node in the conditioning set.

Both directed and undirected graphical models can always be losslessly converted to a factor graph, i.e. such that by applying the above blocking rules after the transformation we obtain exactly the same set of conditional independency properties as present in the original graph, and thus they have a superset of the capacity to represent conditional indendence properties as either of these two alternative frameworks. For example, factor graph equivalents of the directed and undirected graphical model examples in Figure 1.1 are shown in Figure 1.2.

As well as allowing representations of mixed graphs with both directed and undirected factors which cannot be represented with either directed or undirected graphical models, factor graphs are also able to include finer-grained information about the factorisation of the joint

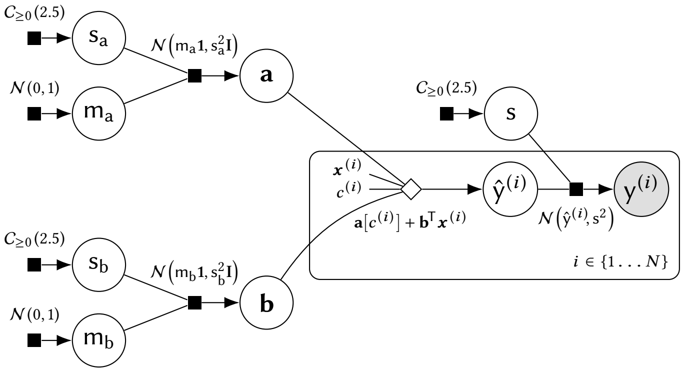

$C_{\geq 0}(2.5)$
$\mathrm{s_a}$
$\mathcal{N}\left(\mathrm{m_a}\mathbf{1}, \mathrm{s_a^2}\mathbf{I}\right)$
$\mathcal{N}(0,1)$
$\mathrm{m_a}$
$\mathbf{a}$
$C_{\geq 0}(2.5)$
s
$\boldsymbol{x}^{(i)}$
$c^{(i)}$
$\hat{\mathrm{y}}^{(i)}$
$\mathrm{y}^{(i)}$
$\mathbf{a}[c^{(i)}] + \mathbf{b}^{\mathsf{T}}\boldsymbol{x}^{(i)}$
$\mathcal{N}\left(\hat{\mathrm{y}}^{(i)}, \mathrm{s}^2\right)$
$i \in \{1 \ldots N\}$
$C_{\geq 0}(2.5)$
$\mathrm{s_b}$
$\mathcal{N}\left(\mathrm{m_b}\mathbf{1}, \mathrm{s_b^2}\mathbf{I}\right)$
$\mathcal{N}(0,1)$
$\mathrm{m_b}$
$\mathbf{b}$

Figure 1.4: Hierarchical linear regression model factor graph showing examples of extended factor graph notation.

density than either of the other two model types by explicitly indicating the presence of individual factors. For instance Figure 1.3 shows the factor graph for a *Boltzmann machine* distribution, sometimes called a *pairwise binary Markov random field* or *Ising model*, on five binary random variables $\{\mathrm{s}_i\}_{i=1}^5$. A Boltzmann machine distribution can be factored in to a product of pairwise weighted interactions $\exp(\mathrm{s}_i W_{ij} \mathrm{s}_j)$ and single variable bias potentials $\exp(b_i \mathrm{s}_i)$, each of which are explicitly represented by labelled factors in Figure 1.3. A corresponding undirected graphical model representation would have a single clique involving all five variables, and so would not indicate any information about the factorisation of the joint density.

In Figure 1.4 we illustrate some additional useful factor graph notation we will use in this thesis. We use a factor graph corresponding to a hierarchical linear regression model which will be discussed in more detail later in the thesis as a motivating example. The exact meaning of the model and its various factors are unimportant to the discussion of notation here so will be skipped for now.

It will often be useful to be able to explicitly represent deterministic functions applied to the random variables in a factor graph. For this purpose we introduce an additional node type denoted by an unfilled diamond (◇). The semantics of this node type are very similar to standard directed factor nodes. Variables acting as inputs to the function are connected to the node by undirected edges and the variable corresponding to the function output indicated by a directed edge from the node to the relevant variable. Like standard factor nodes, the deterministic factor nodes only ever connect to variable nodes. The operations performed by the function on the inputs will usually be included as a label adjacent to the node as illustrated by the example in Figure 1.4.

A deterministic factor node can informally[1] be considered equivalent to a directed factor node with a degenerate Dirac delta conditional density on the output variable which concentrates all the probability mass at the output of the function applied to the inputs variables. The previously discussed rules for evaluating conditional independency properties in factor graphs can be directly extended to account for the new node type by just considering it as a directed factor node.

Optionally constant values used in a model may be included in a factor graph as plain nodes indicated only by a label. The $\boldsymbol{x}^{(i)}$ and $c^{(i)}$ nodes in Figure 1.4 are an example of this notation.

A commonly used convention in factor graphs (and other graphical models) is *plate notation* [17], with an example of a plate shown by the rounded rectangle bounding some of the nodes in Figure 1.4. Plates are used to indicate a subgraph in the model which is replicated multiple times (with the replications being indexed over a set which is typically indicated in the lower right corner of the plate as in Figure 1.4). The subgraph entirely contained on the plate is assumed to be replicated the relevant number of times, with any edges crossing into the plate from variable nodes outside of the plate being repeated once for each subgraph replication. Plates are commonly used to represent a model component repeated across multiple data items.

Each of the factors in Figure 1.4 is labelled with a shorthand for a probability density function corresponding to the conditional or marginal density factor associated with the node. Definitions for the shorthand notations that are used for densities in this thesis are given in Tables 1.2 and 1.3. The dependence of the factors on the value of the random variable the density is defined on is omitted in the labels for brevity.

A final additional notation used in Figure 1.4 is the use of a shaded variable node (corresponding to $\mathrm{y}^{(i)}$) to indicate a random variable corresponding to an observable quantity in the model.

[1] A Dirac delta cannot strictly define a density as it is not the Radon–Nikodyn derivative of an absolutely continuous measure, however it can be informally treated as the density of a singular Dirac measure $f(0) = \int f(x)\,\mathrm{d}\delta(x) \simeq \int f(x)\delta(x)\,\mathrm{d}x$.

1.2.3 Computation graphs

A final graph based tool we will make use of in this theis is that of *computation graphs* [6]. In particular computation graphs (via associated software frameworks [116]) will be used to allow automatic differentiation of complex probabilistic models used in later chapters. Computation graphs are not typically considered in the context of probabilistic graphical models, but they share many of the same features and as we will see are closely related to directed factor graphs.

A *computation graph*, sometimes insead termed a *computational graph* or *data flow graph*, represents the computations involved in evaluating a mathematical expression. In this thesis we will distinguish between two types of nodes in a computation graph. *Variable nodes* correspond to variables which hold either inputs to the computation or intermediate results corresponding to the outputs of sub-expressions. *Operation nodes* describe how non-input variable nodes are computed as functions of other variable nodes. In other presentations of computation graphs often the operation nodes are instead implicitly represented by directed edges between variable nodes. However analagously to the more explicit factorisation afforded by directed factor graphs compared to directed graphical models, directly representing operations as nodes allows finer grained information about the decomposition of the operations associated with a computation graph to be included.

As with directed graphical models and directed factor graphs, computation graphs cannot contain directed cycles. This does not preclude recursive and recurrent computations however as these can always be unrolled to form a directed acyclic graph. The 'mathematical expressions' a computation graph is constructed to evaluate can be arbitarily complex - a computation graph corresponding to the evaluation of any numerical algorithm can always be constructed including use of arbitrary nested flow control and branching statements.

An example of a computation graph representing the calculation of the negative log density of a univariate normal distribution, i.e.

$$c = \frac{1}{2}\left(\frac{x - m}{s}\right)^2 + \log s + \frac{1}{2}\log(2\pi) \tag{1.50}$$

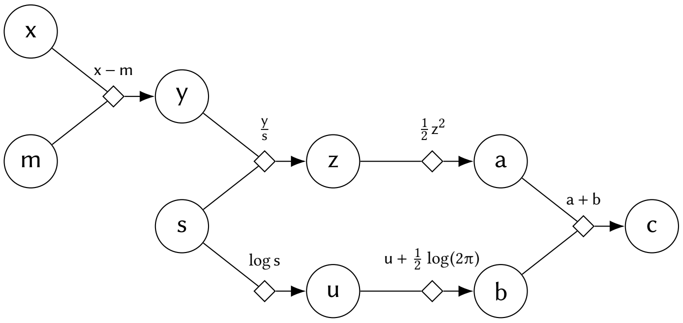


Figure 1.5: Example computation graph corresponding to calculation of the negative log density of a univariate normal distribution.

is shown in Figure 1.5. The graph inputs have chosen to be the value of the random variable (x) to evaluate the density at and the mean (m) and the standard deviation (s) parameters of the density.

Variable nodes in the computation graph have been represented by labelled circles and operation nodes with labelled diamonds. Undirected edges connecting from a variable node to an operation node correspond to the inputs to the operation, and directed edges from an operation node to variable nodes to the outputs of the operation.

The computation graph associated with a given expression is not uniquely defined. There will usually be multiple possible orderings in which operations can be applied to achieve the same result (up to differences due to non-exact floating point computation). Similarly what should considered a single operation to be represented by a node in the computation graph as opposed to being split up into a sub-graph of multiple operations is a matter of choice. For example in Figure 1.5 the addition of the constant $\frac{1}{2}\log(2\pi)$ could have been included at various other points in the graph and the operation $\frac{1}{2}z^2$ could have been split in to separate multiplication and exponentation operations.

1.2.3.1 *Automatic differentiation*

The main motivation for representing expressions as computation graphs is to formalise an efficient general procedure for automatically calculating derivatives of the output of an expression with respect to its inputs termed automatic differentiation [8, 82]. The key ideas in automatic differentiation are to use the chain rule to decompose the derivatives into products and sums of the partial derivatives of the output of each individual operation in the expression with respect to its input, and to

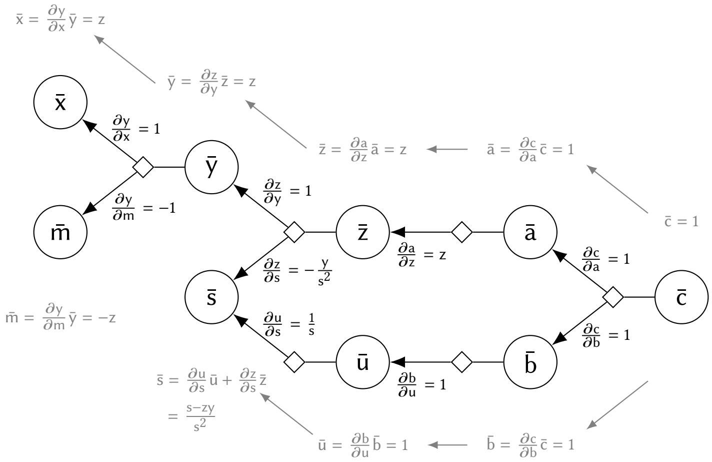

$\bar{x} = \frac{\partial y}{\partial x}\bar{y} = z$
$\bar{y} = \frac{\partial z}{\partial y}\bar{z} = z$
$\bar{x}$
$\frac{\partial y}{\partial x} = 1$
$\bar{y}$
$\bar{z} = \frac{\partial a}{\partial z}\bar{a} = z$
$\bar{a} = \frac{\partial c}{\partial a}\bar{c} = 1$
$\frac{\partial z}{\partial y} = 1$
$\frac{\partial y}{\partial m} = -1$
$\bar{c} = 1$
$\bar{m}$
$\bar{z}$
$\bar{a}$
$\frac{\partial a}{\partial z} = z$
$\frac{\partial c}{\partial a} = 1$
$\frac{\partial z}{\partial s} = -\frac{y}{s^2}$
$\bar{s}$
$\bar{c}$
$\bar{m} = \frac{\partial y}{\partial m}\bar{y} = -z$
$\frac{\partial u}{\partial s} = \frac{1}{s}$
$\frac{\partial c}{\partial b} = 1$
$\bar{u}$
$\bar{b}$
$\bar{s} = \frac{\partial u}{\partial s}\bar{u} + \frac{\partial z}{\partial s}\bar{z}$
$= \frac{s-zy}{s^2}$
$\frac{\partial b}{\partial u} = 1$
$\bar{u} = \frac{\partial b}{\partial u}\bar{b} = 1$
$\bar{b} = \frac{\partial c}{\partial b}\bar{c} = 1$

Figure 1.6: Visualisation of applying reverse-mode automatic differentiation to the computation graph in Figure 1.5 to calculate the derivatives of the negative log density of a univariate normal distribution.

use an efficient recursive accumulation of these partial derivative sum-products corresponding to a traversal of the computation graph such that multiple derivatives can be efficiently calculated together.

Depending on how the computation graph is traversed to accumulate the derivative terms, different modes of automatic differentiation are possible. Of most use in this thesis will be *reverse-mode accumulation* [111], in which the derivatives of an output node with respect to all input nodes are accumulated by a reverse pass through the computation graph from the output node to inputs.

As an example the partial derivatives of the expression for univariate normal log density given in (1.50) with respect to x, m and s can be decomposed using the chain rule in terms of the intermediate variables in the computation graph shown in Figure 1.5 as

$$\frac{\partial c}{\partial x} = \frac{\partial c}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial x}, \tag{1.51}$$

$$\frac{\partial c}{\partial m} = \frac{\partial c}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial y}\frac{\partial y}{\partial m}, \tag{1.52}$$

$$\frac{\partial c}{\partial s} = \frac{\partial c}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial s} + \frac{\partial c}{\partial b}\frac{\partial b}{\partial u}\frac{\partial u}{\partial s}. \tag{1.53}$$

We can immediately see that some of the chains of products of partial derivatives are repeated in the different derivative expressions - for example $\frac{\partial c}{\partial a}\frac{\partial a}{\partial z}$ appears in the expressions for all three derivatives.

Algorithm 1 Reverse-mode automatic differentiation.

Input: $\{x_i\}_{i=1}^{M}$: computation graph input variables,
Pa : indices of parent variables to an operation given its index,
Ch : indices to child operations of a variable given its index,
$\{f_i\}_{i=M+1}^{N}$: computation graph operations in topological order,
$\{\{\partial_j f_i\}_{j \in \mathrm{Pa}(i)}\}_{i=M+1}^{N}$: operation partial derivatives wrt parent variables.
Output: x_N : function output,
$\{\bar{x}_i\}_{i=1}^{N-1}$: partial derivatives of output wrt intermediate and input variables.

```
for $i \in \{M+1 \dots N\}$ do
    $x_i \leftarrow f_i\big(\{x_j\}_{j \in \mathrm{Pa}(i)}\big)$
$\bar{x}_N \leftarrow 1$
for $i \in \{N-1 \dots 1\}$ do
    $\bar{x}_i \leftarrow \sum_{j \in \mathrm{Ch}(i)} \bar{x}_j \partial_i f_j(x_i)$
```

Reverse-mode accumulation is effectively an automatic way of exploiting these possibilities for reusing calculations.

Figure 1.6 shows a visualisation of reverse-mode accumulation applied to the computation graph in Figure 1.5. The first step is for a *forward pass* through the graph to be performed, i.e. values are provided for each of the input variables and then each of the intermediate and output variables calculated from the incoming operation applied to their parent values. Importantly the values of all variables in the graph calculated during the forward pass must be maintained in memory.

The *reverse pass* recursively calculates the values of the partial derivatives of the relevant output node with respect to each variable node in the graph - we will term these intermediate derivatives *accumulators* denoted with barred symbols in Figure 1.6 e.g. $\bar{\mathrm{a}} = \frac{\partial \mathrm{c}}{\partial \mathrm{a}}$. The reverse pass begins by seeding an accumulator for the output node to one (i.e. $\bar{\mathrm{c}} = \frac{\partial \mathrm{c}}{\partial \mathrm{c}} = 1$ in Figure 1.6).

Accumulators for the input variables of an operation are calculated by multiplying the accumulator for the operation output by the partial derivatives of the operation output with respect to each input variable. For non-linear operations multiplying by the operator partial derivatives will require access to the value of the input variables calculated in the forward pass. If a variable is an input to multiple operations, the derivative terms from each operation are added together in the relevant accumulator, as for example shown for $\bar{\mathrm{s}}$ in Figure 1.6. By recursively applying these product and sum operations, the derivatives of the output with respect to all variables in the graph can be calculated. A general description of the method for computation graphs with a single output node and multiple inputs is given in Algorithm 1.

This reverse accumulation method allows computation of numerically exact (up to floating point error) derivatives of a single output variable in a computation graph with respect to *all input variables* with a computational cost, in terms of the number of atomic operations which need to be performed, that is a constant factor of the cost of the evaluation of the original expression represented by the computation graph in the forward pass. The constant factor is typically two to three and at most six [7]. This efficient computational cost is balanced by the requirement that the values of all intermediate variables in the computation graph evaluated in the forward pass through the graph must be stored in memory for the derivative accumulation in a reverse pass, which for large computational graphs can become a bottleneck.

To calculate the full Jacobian from a computation graph representing a function with M inputs $\{\mathsf{x}_i\}_{i=1}^{M}$ and N outputs $\{\mathsf{y}_i\}_{i=1}^{N}$, i.e. the $N \times M$ matrix $\boldsymbol{J}$ with entries $J_{i,j} = \frac{\partial \mathsf{y}_i}{\partial \mathsf{x}_j}$, we can do a single forward pass and N reverse passes each time accumulating the derivatives of one output variable with respect to all inputs. This leads to an overall computational cost that is $\mathcal{O}(N)$ times the cost of a single (forward) function evalaution to evaluate the full Jacobian. As each of the reverse passes can trivially be run in parallel (in addition to any parallelisation of the operations in the forward and reverse passes themselves), this $\mathcal{O}(N)$ factor in the operation count need not corresponds to an equivalent increase in compute time.

An alternative to reverse-mode accumulation is *forward-mode accumulation* [124], which insteads accumulates partial derivatives with respect to a single input variable alongside the forward pass through the graph. In contrast to reverse-mode, this allows calculation of the partial derivatives of all output variables with respect to a single input variable at a computational cost that is a constant factor of the cost of the evaluation of the original expression in the forward pass. Forward-mode accumulation therefore allows evaluation of the Jacobian of a function with M inputs and N outputs at an overall computational cost that is $\mathcal{O}(M)$ times the cost of a single function evaluation.

For functions with $M \gg N$, e.g. scalar valued functions of multiple inputs, reverse-mode accumulation is generally therefore signficantly more efficient at computing the Jacobian. Forward-mode accumulation is however useful for evaluating the Jacobian of functions with $N \gg M$, and also has the advantage over reverse-mode accumulation of avoiding the requirement to store the values of intermediate variables from the forward pass for the reverse pass(es).

The direct overlap in our notation to represent variable and operation nodes in computation graphs and that used to represent (random) variable nodes and deterministic factor nodes in factor graphs is intentional. Although often the operations associated with a deterministic node in a factor graph will be more complex than the operations usually represented by nodes in a computation graph, this is only a matter of granularity of reprensentation - fundamentally they perform the same role. Importantly this means we can treat subgraphs of a factor graphs consisting of only variable and deterministic factor nodes as computation graphs and if the operations performed by the deterministic nodes are differentiable, use reverse-mode automatic differentiation to efficiently propagate derivatives through these sub-graphs.

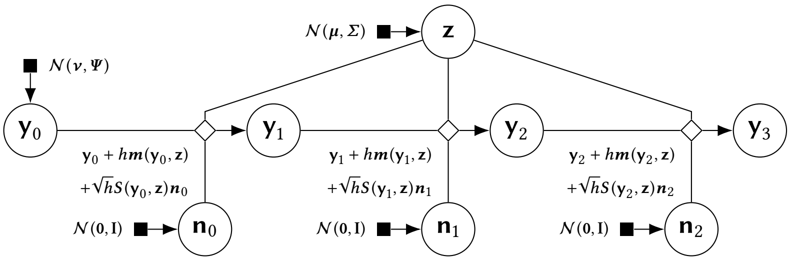
Like directed graphical models, a directed factor graph naturally specifies a generative process via ancestral sampling, with values for the random variables in the graph successively calculated in a forward pass consisting of a combination of deterministic and stochastic operations on the values of parent variables. A computation graph likewise specifies a generative process, how to compute the expression outputs given inputs, computed via a forward pass through the graph with the main differences being here that the inputs to that process are assumed to be given rather than sampled from marginal densities and the intermediate operations are all deterministic.

1.2.4 Simulators

Rather than specifying a generative model via a directed factor graph (or graphical model), it is common for complex models to instead be specified procedurally in code as a *simulator*. Often such simulators may involve a mechanistic model of a physical process for example described by a set of *stochastic differential equations* (SDEs). Any stochasticity in a simulator model will be introduced via draws from a pseudo-random number generator in the programming language used to specify the model. Given these random inputs, the output of the simulator is then calculated as a series of determinstic operations performed to

```
$\mathbf{y}_0 \sim \mathcal{N}(\boldsymbol{\nu}, \boldsymbol{\Psi})$
$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
for $t \in \{1 \dots T\}$ do
    $\mathbf{n}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1} + h\,\boldsymbol{m}(\mathbf{y}_{t-1}, \mathbf{z}) + \sqrt{h}\,\boldsymbol{S}(\mathbf{y}_{t-1}, \mathbf{z})\mathbf{n}_{t-1}$
```

(a) Pseudo-code for Euler–Maruyama simulation of SDE model.

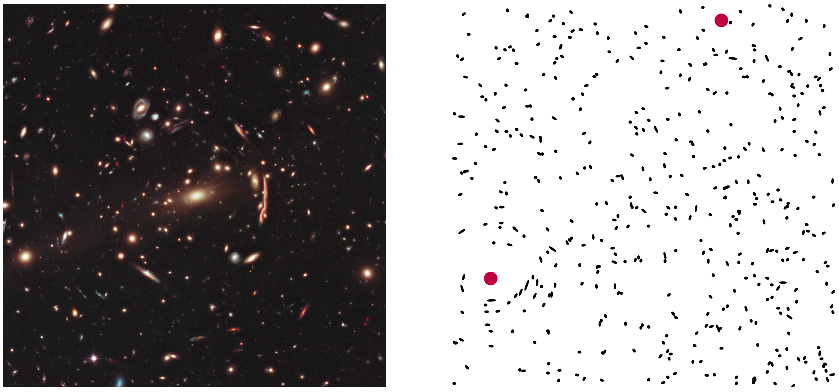

$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\mathbf{z}$
$\mathcal{N}(\boldsymbol{\nu}, \boldsymbol{\Psi})$
$\mathbf{y}_0$
$\mathbf{y}_1$
$\mathbf{y}_2$
$\mathbf{y}_3$
$\mathbf{y}_0 + h\boldsymbol{m}(\mathbf{y}_0, \mathbf{z})$
$+\sqrt{h}\boldsymbol{S}(\mathbf{y}_0, \mathbf{z})\boldsymbol{n}_0$
$\mathbf{y}_1 + h\boldsymbol{m}(\mathbf{y}_1, \mathbf{z})$
$+\sqrt{h}\boldsymbol{S}(\mathbf{y}_1, \mathbf{z})\boldsymbol{n}_1$
$\mathbf{y}_2 + h\boldsymbol{m}(\mathbf{y}_2, \mathbf{z})$
$+\sqrt{h}\boldsymbol{S}(\mathbf{y}_2, \mathbf{z})\boldsymbol{n}_2$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{n}_0$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{n}_1$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{n}_2$

(b) Directed factor graph of 3 time steps of SDE simulation.

Figure 1.7: Example of a simulator model corresponding to Euler–Maruyma integration of a set of *stochastic differential equations* (SDEs), $\mathrm{d}\mathbf{y}(t) = \boldsymbol{m}(\mathbf{y}(t), \mathbf{z})\,\mathrm{d}t + \boldsymbol{S}(\mathbf{y}(t), \mathbf{z})\,\mathrm{d}\mathbf{n}(t)$, specified as pseudo-code in (a) and a directed factor graph in (b). In the pseudo-code the notation $\sim$ followed by a distribution shorthand represents generating a value from the associated distribution and assigning it to a variable.

the inputs and so can be described by a computation graph. The overall composition of directed factor nodes specifying the generation of random inputs from known densities by the random number generator and computation graph describing the operations performed by the simulator code together therefore define a directed factor graph from which we can extract a joint density on all the variables in the models as the product of all factors (with implicit Dirac delta terms on the outputs of deterministic factors). An example of a simulator model corresponding to Euler–Maruyma approximate integration [57] of a set of SDEs is shown as both pseudo-code and a factor graph in Figure 1.7.

A key difference of simulator models from more typical probabilistic models is that the variables corresponding to observables in the factor graph of a simulator model may be the output of deterministic factors rather than probabilistic directed factors. As we will see later in the thesis this can complicate inference in such models.



(a) Hubble Space Telescope image of galaxy cluster MACS J1206 showing visible distortion due to gravitational lensing. Image credit: ESA/Hubble.

(b) Simulated galaxy cluster image from *Observing Dark Worlds* data [52] showing ellipticity of galaxies (black ellipses) distorted by dark matter halos (red circles).

Figure 1.8: Gravitational lensing in real and simulated galaxy cluster images.

1.3 INFERENCE

Having now introduced the tools we use to construct probabilistic models, we will now describe more concretely the task of inference. To help motivate our exposition we will discuss inference in the context of a specific problem: inferring the location of dark matter halos from the observed gravitational lensing of light emitted by surrounding galaxies. This task was the inspiration for a Kaggle[2] competition, *Observing Dark Worlds* [45, 52] and we will use the simplified formulation of the task used in the competition as the basis for our discussion here.

1.3.1 Example problem: Observing Dark Worlds

We will begin with a brief review of the motivation for the *Observing Dark Worlds* inference problem. Dark matter is a hypothesised form of matter which does not emit or interact with electromagnetic radiation. This prevents direct observation of dark matter by telescopes as it produces no signal in any part of the electromagnetic spectrum [70]. General relativity however predicts that all objects with mass locally distort spacetime. If a very large concentration of mass lies between an observer and a light source the strong distortion of spacetime by the mass significantly alters the paths of photons emitted by the source and causes the apparent image of the source to the observer to be vis-

[2] An online platform for predictive modelling competitions https://www.kaggle.com.

ibly distorted [5]. This effect is analagous to placing a lens between the light source and observer and this motivates the term *gravitational lensing* to describe the phenonemon. It is believed that large concentrations of dark matter around galaxy clusters termed *dark matter halos*, cause gravitational lensing of the light emitted from background galaxies observed in telescope imagery of galaxy clusters.

Galaxies typically have approximately elliptical shapes in telescope images. It is assumed that the *ellipticities* of observed galaxy images not subject to gravitational lensing are isotropically distributed [5]: there is no preferred orientation of the galaxies to an observer on Earth. The presence of a large mass concentration such as a dark matter halo between background galaxies and a telescope however locally distorts the distribution of the ellipticities of the observed galaxy images. An example of this effect in a galaxy cluster image from the Hubble Space Telescope is shown in Figure 1.8a with the galaxies showing a bias towards being oriented tangentially to the bright region at the centre of the image. Local biases in the spatial distribution of galaxy ellipticities can therefore be used to infer the location of dark matter halos.

In the context of gravitational lensing ellipticity is defined as a complex quantity $\epsilon = \frac{a-b}{a+b}\exp(2i\phi)$ where a is the length of the ellipse major axis, b the minor axis length and ϕ the orientation of the major axis. Here we will parameterise ellipticity terms of $e_1 = \mathrm{Re}(\epsilon)$ and $e_2 = \mathrm{Im}(\epsilon)$.

The *Observing Dark Worlds* inference task is formulated as follows. The observed data consists of the sky co-ordinates $\{u_g, v_g\}_{g=1}^{G}$ and ellipticity components $\{e_{1,g}, e_{2,g}\}_{g=1}^{G}$ of a set of G galaxies. The task is given this data to infer the coordinates $\{\mathsf{x}_h, \mathsf{y}_h\}_{h=1}^{H}$ of the centers of a known number of dark matter halos present in the sky field of view. Galaxy coordinate and ellipticity data are provided for multiple cluster images each with a known number of halos present and the cluster image data are assumed to be *independently and identically distributed* (iid).

The *Observing Dark Worlds* data is simulated thus the ground truth positions of the dark matter halos are known in reality which was required for the purposes of evaluation in the competition. An example visualisation of one of the simulated galaxy cluster images in the data set is shown in Figure 1.8b. The known positions of the dark matter halos in this image are shown by red circles[3]. As can be seen the halo in the lower left of this image produces a strong visible distortion in the ellipt-

[3] A *training set* of cluster image data was provided to competition participants for which the associated true dark matter halo positions were given. This data was distinct from the *test set* cluster data were the halo positions are unknown and which is the basis of the described inference problem and scoring for the competition, with the training data intended to aid initial model exploration and evaluation.

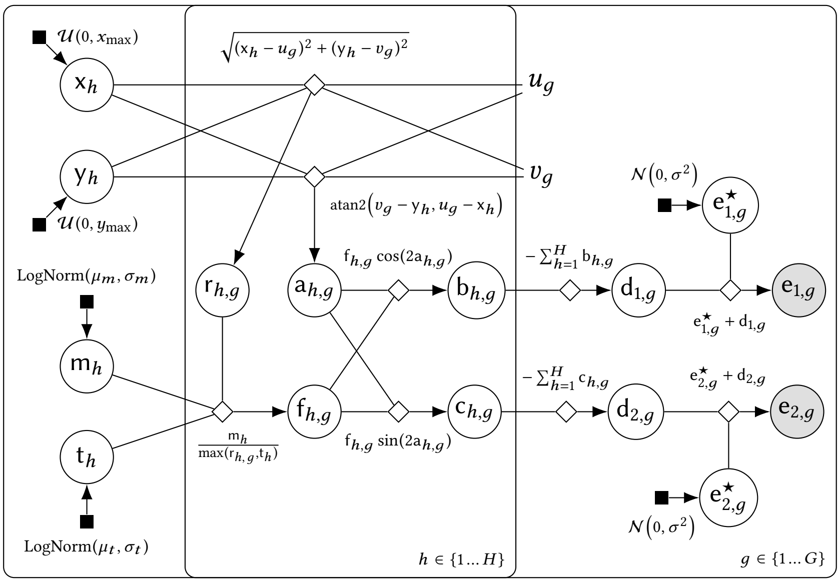

$\mathcal{U}(0, x_{\max})$
x_h
$\sqrt{(\mathrm{x}_h - u_g)^2 + (\mathrm{y}_h - v_g)^2}$
u_g
y_h
v_g
$\mathcal{U}(0, y_{\max})$
$\mathrm{atan2}\big(v_g - \mathrm{y}_h, u_g - \mathrm{x}_h\big)$
$\mathcal{N}\big(0, \sigma^2\big)$
$\mathrm{e}^{\star}_{1,g}$
LogNorm(μ_m, σ_m)
$\mathrm{r}_{h,g}$
$\mathrm{a}_{h,g}$
$\mathrm{f}_{h,g}\cos(2\mathrm{a}_{h,g})$
$\mathrm{b}_{h,g}$
$-\sum_{h=1}^{H}\mathrm{b}_{h,g}$
$\mathrm{d}_{1,g}$
$\mathrm{e}_{1,g}$
$\mathrm{e}^{\star}_{1,g} + \mathrm{d}_{1,g}$
m_h
$-\sum_{h=1}^{H}\mathrm{c}_{h,g}$
$\mathrm{e}^{\star}_{2,g} + \mathrm{d}_{2,g}$
$\mathrm{f}_{h,g}$
$\mathrm{c}_{h,g}$
$\mathrm{d}_{2,g}$
$\mathrm{e}_{2,g}$
t_h
$\frac{\mathrm{m}_h}{\max(\mathrm{r}_{h,g}, \mathrm{t}_h)}$
$\mathrm{f}_{h,g}\sin(2\mathrm{a}_{h,g})$
$\mathrm{e}^{\star}_{2,g}$
$\mathcal{N}\big(0, \sigma^2\big)$
LogNorm(μ_t, σ_t)
$h \in \{1 \ldots H\}$
$g \in \{1 \ldots G\}$

Figure 1.9: Factor graph of proposed model for *Observing Dark Worlds* gravitational lensing inference problem for a single cluster image.

icities, however the second halo at the top right has a much less visible effect on the surrounding galaxies.

The use of simulated data reduces the difficulty of the *Observing Dark Worlds* inference task compared to working with real gravitational lensing data, however for the purposes of our discussion the task is still sufficiently complex to highlight the computational challenges involved in inference problems. Further the probabilistic approach described here is similar to that used (albeit with significantly more complex models) in methods and tools used in practice to analyse gravitational lensing data to infer dark matter properties [51, 69].

1.3.2 Defining a model

You cannot do inference without making assumptions —David Mackay

Our starting point for tackling inference problems will be to define a probabilistic model specifying proposed relationships between the observed and non-observed quantities to be inferred. The model codifies the assumptions we make about the problem and any prior beliefs we have. In virtually all real inference problems the model will be a significantly simplified description of a much more complex underlying process, usually motivated by prior empirical observations that the behaviour proposed by the model is a reasonable description of reality. For now we will consider the model as a singular fixed object we will perform inference with. In reality probabilistic modelling and inference are an iterative process with model criticism a key part of the loop [15, 34]. We will discuss some of the (computational) issues involved in probabilistic model evaluation and comparison at the end of this chapter.

For the *Observing Dark Worlds* problem, a proposed probabilistic model is shown as a directed factor graph in Figure 1.9[4]. This model assumes a simple, physically motivated relationship between the dark matter halo positions and the observed spatial distribution of galaxy ellipticities. As well as random variables for the halo coordinates $\{\mathsf{x}_h, \mathsf{y}_h\}_{h=1}^{H}$ and galaxy ellipticity components $\{\mathsf{e}_{1,g}, \mathsf{e}_{2,g}\}_{g=1}^{G}$, the model also introduces additional latent (non-observed) random variables $\{\mathsf{m}_h, \mathsf{t}_h\}_{h=1}^{H}$, which correpond respectively to the unknown total masses of each halo (multiplied by an arbitrary scaling factor) and an unknown *core radius* for each halo, which specifies a radial distance within which the distortion by the halo is of constant magnitude. The halo coordinates are assumed to be marginally uniformly distributed across the image extents. The halo mass and core radius latent variables are both positive quanties and are assumed to have log normal marginal distributions.

The model proposes that at sky coordinate (u, v) each halo produces a shear distortion of magnitude

$$f_h(u, v) = \frac{\mathsf{m}_h}{\max(r_h, \mathsf{t}_h)} \quad \text{where} \quad r_h = \sqrt{(\mathsf{x}_h - u)^2 + (\mathsf{y}_h - v)^2} \tag{1.54}$$

and acting in a tangential direction to the radial vector from the halo centre $(\mathsf{x}_h, \mathsf{y}_h)$ to (u, v). This functional relationship is just one possibility among many. The post-competition review article [45] revealed that the simulated data actually used dark matter halos with a mixture of two different radial density profiles, neither of which correspond to the distortion model assumed in (1.54). This mismatch between a model and the actual process by which observations were generated will virtually always be the case in real inference problems. We can still make inferences which are consistent with our modelling assumptions, however we should as far as possible also critically review those modelling assumptions to check the validity of the inferences made.

For galaxies where the variation of the magnitude of the distorting effect of a halo across the extent of the galaxy's image is small, a reason-

[4] The factor graph in Figure 1.9 is based on the models used by the participants with the top-two winning entries in the Kaggle competition, Iain Murray and Tim Salimans, and described in their personal reports on their competition entries [79, 104] and an article evaluating the competition outcomes [45].

able approximation of the gravitational lensing effect of a single halo on the observed ellipticity $(\mathrm{e}_{1,g}, \mathrm{e}_{2,g})$ of a galaxy image with intrinsic (prior to any gravitational lensing effect) ellipticity $(\mathrm{e}^{\star}_{1_g}, \mathrm{e}^{\star}_{2})$ is that

$$\mathrm{e}_{1,g} = \mathrm{e}^{\star}_{1,g} - \mathrm{f}_{h,g}\cos(2\mathrm{a}_{h,g}), \quad \mathrm{e}_{2,g} = \mathrm{e}^{\star}_{2,g} - \mathrm{f}_{h,g}\sin(2\mathrm{a}_{h,g}), \tag{1.55}$$

where $\mathrm{a}_{h,g} = \operatorname{atan2}\left(v_g - \mathrm{y}_h, u_g - \mathrm{x}_g\right)$ is the angle the line from the centre of halo h to galaxy g makes to the horizontal axis and $\mathrm{f}_{h,g} = f_h(u_g, v_g)$ is the magnitude of the shear distortion according to the proposed relationship in Equation (1.54) evaluated at the galaxy image centre [5, 69]. For clusters with multiple halos, a further simple linearity assumption is made, that the shear distortions due to the different halos act additively on the ellipticity components

$$\mathrm{e}_{1,g} = \mathrm{e}^{\star}_{1,g} - \sum_{h=1}^{H} \mathrm{f}_{h,g}\cos(2\mathrm{a}_{h,g}), \ \mathrm{e}_{2,g} = \mathrm{e}^{\star}_{2,g} - \sum_{h=1}^{H} \mathrm{f}_{h,g}\sin(2\mathrm{a}_{h,g}). \tag{1.56}$$

The intrinsic ellipticities of the galaxies are assumed to have a isotropic, zero-mean normal distribution, with standard deviation σ, with normal assumption corresponding well to empirical observations from large scale surveys [5, 69]. For now we will assume the standard deviation σ of the intrinsic ellipticities, and the parameters μ_m, σ_m, μ_t, σ_t of the halo mass and core radius marginal distributions have somehow been set to reasonable values, for example based on prior beliefs about the typical ranges of the variables. We will discuss possible strategies for choosing (or inferring) these parameters in more detail later.

1.3.3 Making predictions

Having now defined a model for the *Observing Dark Worlds* problem, we now consider how to use this model to make predictions about the unobserved dark matter halo positions. The downstream task we are using the inference output for will generally determine what the exact inferential query we wish to evaluate is. In general however, any prediction output which takes into account all of the information we have about the unobserved variables given the assumed model and observed data will be computed as a conditional expectation.

To motivate this statement for the *Observing Dark Worlds* example we will discuss some instances of outputs we might wish to compute. For notational convenience we define the following random vectors for the halo random variables

$$\mathsf{x} = \begin{bmatrix} \mathsf{x}_1 \\ \vdots \\ \mathsf{x}_H \end{bmatrix}, \quad \mathsf{y} = \begin{bmatrix} \mathsf{y}_1 \\ \vdots \\ \mathsf{y}_H \end{bmatrix}, \quad \mathsf{m} = \begin{bmatrix} \mathsf{m}_1 \\ \vdots \\ \mathsf{m}_H \end{bmatrix}, \quad \mathsf{t} = \begin{bmatrix} \mathsf{t}_1 \\ \vdots \\ \mathsf{t}_H \end{bmatrix}, \tag{1.57}$$

and similarly for the observed and intrinsic galaxy ellipticities

$$\mathsf{e}_1 = \begin{bmatrix} \mathsf{e}_{1,1} \\ \vdots \\ \mathsf{e}_{1,G} \end{bmatrix}, \quad \mathsf{e}_2 = \begin{bmatrix} \mathsf{e}_{2,1} \\ \vdots \\ \mathsf{e}_{2,G} \end{bmatrix}, \quad \mathsf{e}_1^\star = \begin{bmatrix} \mathsf{e}_{1,1}^\star \\ \vdots \\ \mathsf{e}_{1,G}^\star \end{bmatrix}, \quad \mathsf{e}_2^\star = \begin{bmatrix} \mathsf{e}_{2,1}^\star \\ \vdots \\ \mathsf{e}_{2,G}^\star \end{bmatrix}. \tag{1.58}$$

An obvious output we may wish to compute is the expected (mean) values of the halo positions given the observed ellipticities. From a decision theoretic standpoint these values correspond to the position predictions which minimise the expected squared error loss from the true positions given the model and observed data. As conditonal expectations these are simply

$$\boldsymbol{m}_{\mathsf{x}} = \mathbb{E}[\mathsf{x} \mid \mathsf{e}_1 = \boldsymbol{e}_1,\, \mathsf{e}_2 = \boldsymbol{e}_2] \text{ and } \boldsymbol{m}_{\mathsf{y}} = \mathbb{E}[\mathsf{y} \mid \mathsf{e}_1 = \boldsymbol{e}_1,\, \mathsf{e}_2 = \boldsymbol{e}_2]. \tag{1.59}$$

We may also be interested in the covariances of the halo positions conditioned on the observations, these giving some idea of our remaining uncertainty in the positions and any correlations between them after conditioning on the observed ellipticities. Again these can be expressed as conditional expectations

$$\begin{aligned} C_{\mathsf{x},\mathsf{x}} &= \mathbb{E}\Big[(\mathsf{x} - \boldsymbol{m}_{\mathsf{x}})(\mathsf{x} - \boldsymbol{m}_{\mathsf{x}})^{\mathsf{T}} \mid \mathsf{e}_1 = \boldsymbol{e}_1,\, \mathsf{e}_2 = \boldsymbol{e}_2\Big], \\ C_{\mathsf{y},\mathsf{y}} &= \mathbb{E}\Big[(\mathsf{y} - \boldsymbol{m}_{\mathsf{y}})(\mathsf{y} - \boldsymbol{m}_{\mathsf{y}})^{\mathsf{T}} \mid \mathsf{e}_1 = \boldsymbol{e}_1,\, \mathsf{e}_2 = \boldsymbol{e}_2\Big], \\ \text{and} \quad C_{\mathsf{x},\mathsf{y}} &= \mathbb{E}\Big[(\mathsf{x} - \boldsymbol{m}_{\mathsf{x}})(\mathsf{y} - \boldsymbol{m}_{\mathsf{y}})^{\mathsf{T}} \mid \mathsf{e}_1 = \boldsymbol{e}_1,\, \mathsf{e}_2 = \boldsymbol{e}_2\Big]. \end{aligned} \tag{1.60}$$

In the *Observing Dark Worlds* competition, participants' halo position predictions were evaluated by comparing to the known true halo positions using a metric provided as part of the competition instructions. If we denote the metric $\ell(\boldsymbol{x}, \boldsymbol{y}; \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ where $(\boldsymbol{x}, \boldsymbol{y})$ are the true halo positions and $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ the predicted positions, then the optimal predictions given the assumed model and observed data can be calculated by minimising the expected metric value conditioned on the observed data

$$\bar{\ell}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \mathbb{E}\big[\ell(\mathbf{x}, \mathbf{y}; \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \mid \mathbf{e}_1 = \boldsymbol{e}_1, \mathbf{e}_2 = \boldsymbol{e}_2\big]. \tag{1.61}$$

Due to linearity of the expectation operator, we can also calculate the derivatives of the expected metric with respect to the predictions as

$$\begin{aligned} &\frac{\partial \bar{\ell}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{x}}} = \mathbb{E}\left[\frac{\partial \ell(\mathbf{x}, \mathbf{y}; \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{x}}} \,\middle|\, \mathbf{e}_1 = \boldsymbol{e}_1, \mathbf{e}_2 = \boldsymbol{e}_2\right] \\ \text{and}\quad &\frac{\partial \bar{\ell}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{y}}} = \mathbb{E}\left[\frac{\partial \ell(\mathbf{x}, \mathbf{y}; \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})}{\partial \hat{\boldsymbol{y}}} \,\middle|\, \mathbf{e}_1 = \boldsymbol{e}_1, \mathbf{e}_2 = \boldsymbol{e}_2\right]. \end{aligned} \tag{1.62}$$

For some metrics we will be able to analytically solve for the stationary points to find the predictions minimising $\bar{\ell}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ and more generally we can use the derivatives in an iterative optimisation scheme.

Evaluating conditional expectations of functions of the unobserved variables in a model given observed data is therefore the key computational task in making inferences about the variables in a probabilistic model. The *Observing Dark Worlds* model factor graph in Figure 1.9 defines a joint density across all the variables in the model. We can therefore use Equation (1.43) to write the conditional expectation of a measurable function f of the halo position random variables $\mathbf{x}$ and $\mathbf{y}$ as[5]

$$\begin{aligned} &\mathbb{E}\big[f(\mathbf{x}, \mathbf{y}) \mid \mathbf{e}_1 = \boldsymbol{e}_1, \mathbf{e}_2 = \boldsymbol{e}_2\big] = \\ &\qquad \iint f(\boldsymbol{x}, \boldsymbol{y})\, \mathrm{p}_{\mathbf{x},\mathbf{y}|\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{e}_1, \boldsymbol{e}_2)\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}, \end{aligned} \tag{1.63}$$

where the conditional density $\mathrm{p}_{\mathbf{x},\mathbf{y}|\mathbf{e}_1,\mathbf{e}_2}$ is defined as

$$\mathrm{p}_{\mathbf{x},\mathbf{y}|\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{e}_1, \boldsymbol{e}_2) = \frac{\mathrm{p}_{\mathbf{x},\boldsymbol{y},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}_1, \boldsymbol{e}_2)}{\mathrm{p}_{\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{e}_1, \boldsymbol{e}_2)}, \tag{1.64}$$

with $\mathrm{p}_{\mathbf{e}_1,\mathbf{e}_2}$ defined in terms of $\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{e}_1,\mathbf{e}_2}$ by marginalising out $\mathbf{x}$ and $\mathbf{y}$

$$\mathrm{p}_{\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{e}_1, \boldsymbol{e}_2) = \iint \mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{e}_1, \boldsymbol{e}_2)\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y}. \tag{1.65}$$

[5] For brevity the sets on which integrals are evaluated have been omitted in this section and should be assumed to be the full co-domain of the corresponding random vector.

We can express $\mathrm{p}_{\mathbf{x},\boldsymbol{y},\mathbf{e}_1,\mathbf{e}_2}$ itself as a marginal of the joint density over all random variables in the model (which we can read off as a product of factors from Figure 1.9) by marginalising out $\mathbf{m}$, $\mathbf{t}$, $\mathbf{e}_1^\star$ and $\mathbf{e}_2^\star$

$$\begin{aligned}\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e}_1,\boldsymbol{e}_2) = \iiiint & \mathrm{p}_{\mathbf{x}}(\boldsymbol{x})\,\mathrm{p}_{\mathbf{y}}(\boldsymbol{y})\,\mathrm{p}_{\mathbf{m}}(\boldsymbol{m})\,\mathrm{p}_{\mathbf{t}}(\boldsymbol{t}) \\ & \mathrm{p}_{\mathbf{e}_1\,|\,\mathbf{e}_1^\star,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}(\boldsymbol{e}_1\,|\,\boldsymbol{e}_1^\star,\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}) \\ & \mathrm{p}_{\mathbf{e}_2\,|\,\mathbf{e}_2^\star,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}(\boldsymbol{e}_2\,|\,\boldsymbol{e}_2^\star,\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}) \\ & \mathrm{p}_{\mathbf{e}_1^\star}(\boldsymbol{e}_1^\star)\,\mathrm{p}_{\mathbf{e}_2^\star}(\boldsymbol{e}_2^\star)\,\mathrm{d}\boldsymbol{e}_1^\star\mathrm{d}\boldsymbol{e}_2^\star\mathrm{d}\boldsymbol{t}\,\mathrm{d}\boldsymbol{m}.\end{aligned} \tag{1.66}$$

The integration with respect to $\boldsymbol{e}_1^\star$ and $\boldsymbol{e}_2^\star$ can be performed analytically as the conditional density factors $\mathrm{p}_{\mathbf{e}_2\,|\,\mathbf{e}_2^\star,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}$ and $\mathrm{p}_{\mathbf{e}_2\,|\,\mathbf{e}_1^\star,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}$ correspond to deterministic factors which depend linearly on $\mathbf{e}_1^\star$ and $\mathbf{e}_2^\star$. Expressing the deterministic factors as Dirac deltas and denoting the functions corresponding to the computational graphs in Figure 1.9 mapping from $\{\mathrm{x}_h,\mathrm{y}_h,\mathrm{m}_h,\mathrm{t}_h\}_{h=1}^H$ to $\{\mathrm{d}_{1,g}\}_{g=1}^G$ and $\{\mathrm{d}_{2,g}\}_{g=1}^G$, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ respectively, we have that

$$\begin{aligned}&\mathrm{p}_{\mathbf{e}_1\,|\,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}(\boldsymbol{e}_1\,|\,\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}) \\ &\quad= \int \mathrm{p}_{\mathbf{e}_1\,|\,\mathbf{e}_1^\star,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}(\boldsymbol{e}_1\,|\,\boldsymbol{e}_1^\star,\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t})\,\mathrm{p}_{\mathbf{e}_1^\star}(\boldsymbol{e}_1^\star)\,\mathrm{d}\boldsymbol{e}_1^\star \\ &\quad= \int \delta\big(\boldsymbol{e}_1-\boldsymbol{e}_1^\star-\boldsymbol{d}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t})\big)\,\mathcal{N}\big(\boldsymbol{e}_1^\star\,|\,\mathbf{0},\sigma^2\mathbf{I}\big)\,\mathrm{d}\boldsymbol{e}_1^\star \\ &\quad= \mathcal{N}\big(\boldsymbol{e}_1\,|\,\boldsymbol{d}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}),\sigma^2\mathbf{I}\big),\end{aligned} \tag{1.67}$$

and likewise for $\mathrm{p}_{\mathbf{e}_2\,|\,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}$

$$\mathrm{p}_{\mathbf{e}_2\,|\,\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t}}(\boldsymbol{e}_2\,|\,\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}) = \mathcal{N}\big(\boldsymbol{e}_2\,|\,\boldsymbol{d}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}),\sigma^2\mathbf{I}\big). \tag{1.68}$$

Rewriting (1.66) in terms of (1.67) and (1.68) and substituting the definitions for the factors from Figure 1.9 we have that

$$\begin{aligned}\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{e}_1,\boldsymbol{e}_2) &= \iint \mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t},\boldsymbol{e}_1,\boldsymbol{e}_2)\,\mathrm{d}\boldsymbol{t}\,\mathrm{d}\boldsymbol{m} \\ = \iint \prod_{h=1}^{H} & \big(\mathcal{U}(x_h\,|\,0,x_{\max})\,\mathcal{U}(y_h\,|\,0,y_{\max})\big) \\ & \mathcal{N}\big(\boldsymbol{e}_1\,|\,\boldsymbol{d}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}),\sigma^2\mathbf{I}\big)\,\mathcal{N}\big(\boldsymbol{e}_2\,|\,\boldsymbol{d}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t}),\sigma^2\mathbf{I}\big) \\ & \mathrm{LogNorm}\big(\boldsymbol{m}\,|\,\mu_m\mathbf{1},\sigma_m^2\mathbf{I}\big)\,\mathrm{LogNorm}\big(\boldsymbol{t}\,|\,\mu_m\mathbf{1},\sigma_m^2\mathbf{I}\big)\,\mathrm{d}\boldsymbol{t}\,\mathrm{d}\boldsymbol{m}.\end{aligned} \tag{1.69}$$

We cannot simplify this integral any further analytically. Thefore the conditional expectation we wish to compute in terms of integrals of functions we can evaluate pointwise is

$$\mathbb{E}[f(\mathbf{x},\mathbf{y}) \mid \mathbf{e}_1 = \boldsymbol{e}_1, \mathbf{e}_2 = \boldsymbol{e}_2] = \frac{\iiiint f(\boldsymbol{x},\boldsymbol{y})\, \mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t},\boldsymbol{e}_1,\boldsymbol{e}_2)\,\mathrm{d}\boldsymbol{t}\,\mathrm{d}\boldsymbol{m}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}}{\iiiint \mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{m},\mathbf{t},\mathbf{e}_1,\mathbf{e}_2}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t},\boldsymbol{e}_1,\boldsymbol{e}_2)\,\mathrm{d}\boldsymbol{t}\,\mathrm{d}\boldsymbol{m}\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}}. \tag{1.70}$$

Each of the vector integration variables in (1.70) is of dimension H and so the overall dimension of the space being integrated over in both the numerator and denominator is $4H$. In the *Observing Dark Worlds* data the number of halos per cluster image H is between one and three so to evaluate conditional expectations of the halo positions we need to compute integrals over spaces with four, eight or twelve dimensions.

For four or eight dimensions it may be feasible to use quadrature methods [23], which involve evaluating the integrand across a fixed grid of points and then computing a weighted sum of these values, to numerically approximate the integrals to reasonable accuracy. For a fixed grid resolution however the cost of quadrature scales exponentially with the dimension of the space being integrated over - if N points are used per dimension, using quadrature to evaluate (1.70) will involve N^{4H} evaluations of the integrand.

Assuming the computational cost of the function $f(\mathbf{x},\mathbf{y})$ the conditional expectation is being calculated of is negligible, the dominant cost in the evaluation of the integrands in (1.70) will be in evaluating $\boldsymbol{d}_1(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t})$ and $\boldsymbol{d}_2(\boldsymbol{x},\boldsymbol{y},\boldsymbol{m},\boldsymbol{t})$ which involve computing the distances and angles between all galaxy-halo pairs. Using a very simplistic (and conservative) assumption that each arithmetic, square root, comparison and trigonometric operation has a unit floating point operation cost, the computation graph in Figure 1.9 which is present on the overlap between the two plates will involve 15 floating point operations in a single forward pass and is evaluated GH times for an overall cost per integrand evaluation of $15GH$ floating point operations.

A rough but conservative lower bound on the floating point operation count of evaluating the integrals in (1.70) using quadrature is therefore $15GHN^{4H}$. The number of galaxies G per cluster image in the *Observing Dark Worlds* data varies between 300 and 740. If we assume $G = 500$

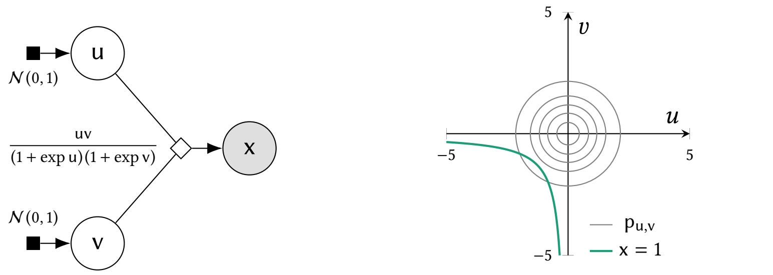
and that we use $N = 20$ grid points per dimension, for the $H = 2$ cases evaluating the conditional expectation will involve $\sim 1 \times 10^9$ floating point operations, for $H = 2$, $\sim 4 \times 10^{14}$ floating point operations and for $H = 3$, $\sim 9 \times 10^{19}$ floating point operations.

At the time of writing, the theoretical peak floating point operation performance of a top-end multi-core server *central processing unit* (CPU) is around 5×10^{11} *floating point operations per second* (FLOPS). Assuming this peak performance could be obtained when using quadrature to approximate (1.70) , our back-of-the-envelope calculation suggests a trivial two millisecond compute time for clusters with one halo, a more noticeable thirteen minute computation for clusters with two halos, and an impractical six year wait for clusters with three halos.

Current top-end *graphics processing units* (GPUs) have a peak theoretical (single-precision) floating point performance of around 10^{13} FLOPS which at best would cut the computation time for a $H = 3$ case by a factor of 20 to around 100 days. With a cluster of CPU or GPU nodes, or faster individual nodes due to future growth in processing power, the compute time could potentially be brought down further to more reasonable timescales. However the key point in this example is the exponential growth with dimension - even moving from eight to twelve dimensions made compute time impractical. Clearly therefore approximating the integrals in expectations like (1.70) using quadrature methods is not viable when performing inference in models with large numbers of unobserved variables, with the example here showing that even integrals over what might initially seem a low dimensionality of twelve can be problematic.

1.3.4 Implicit models

In the *Observing Dark Worlds* example it was possible to analytically integrate out the random vectors $\mathbf{e}_1^\star$ and $\mathbf{e}_2^\star$, which correspond to the intrinsic galaxy ellipticities, due to the observed ellipticities $\mathbf{e}_1$ and $\mathbf{e}_2$ being deterministic linear functions of $\mathbf{e}_1^\star$ and $\mathbf{e}_2^\star$ respectively (for fixed $\mathbf{x}$, $\mathbf{y}$, $\mathbf{m}$ and $\mathbf{t}$). Rather than calculating the conditional densities on $\mathbf{e}_1$ and $\mathbf{e}_2$ by integrating out $\mathbf{e}_1^\star$ and $\mathbf{e}_2^\star$, we could equivalently have applied the change of variables formula (1.28) for a bijective transformation; in this case the Jacobian determinant is simply one.



(a) Directed factor graph defining model with two latent variables (u, v), and an observed variable x.

(b) Plot of density on latent variables (contours) and set of values for which x = 1 (green curve).

Figure 1.10: Simple example of an implicit probabilistic model where the observed variable is a non-bijective function of two latent variables.

If the observed variables had instead been the output of a deterministic factor where there was no bijective dependence on any of the parent variables (inputs to the deterministic factor), the simplified form of the change of variables formula (1.28) would no longer have applied and instead the more general form (1.26) would have been required. Generally in such cases it will not be possible to analytically marginalise out a parent variable to give an explicit conditional density on the observed variable(s). Models which do not admit explicit conditional densities on the observed variables are sometimes described as *implicit models* [26], with simulator models being a common case.

An illustration of such a case for a simple three variable model is shown in Figure 1.10. Here the observed variable x is a deterministic function of two latent (unobserved) variables u and v. There is no analytic solution in terms of elementary functions for u as a function of x and v or for v as a function of x and u. This means the Dirac delta term corresponding to the deterministic factor cannot be integrated out. Due to the presence of the Dirac delta the joint density $p_{x,u,v}$ is not well defined (the joint probability $P_{x,u,m}$ is not absolutely continuous with respect to any measure) which complicates evaluations of conditional expectations such as $\mathbb{E}[f(u,v) \mid x = 1]$.

In particular the set of u and v values corresponding to solutions to $x = 1$ (illustrated as the green curve in Figure 1.1a) has zero Lebesgue measure. Therefore even though the dimensionality is low in this case we can not use simple quadrature to evaluate conditional expectations without some further form of approximation. We will revisit methods for performing inference in implicit models later in the thesis.

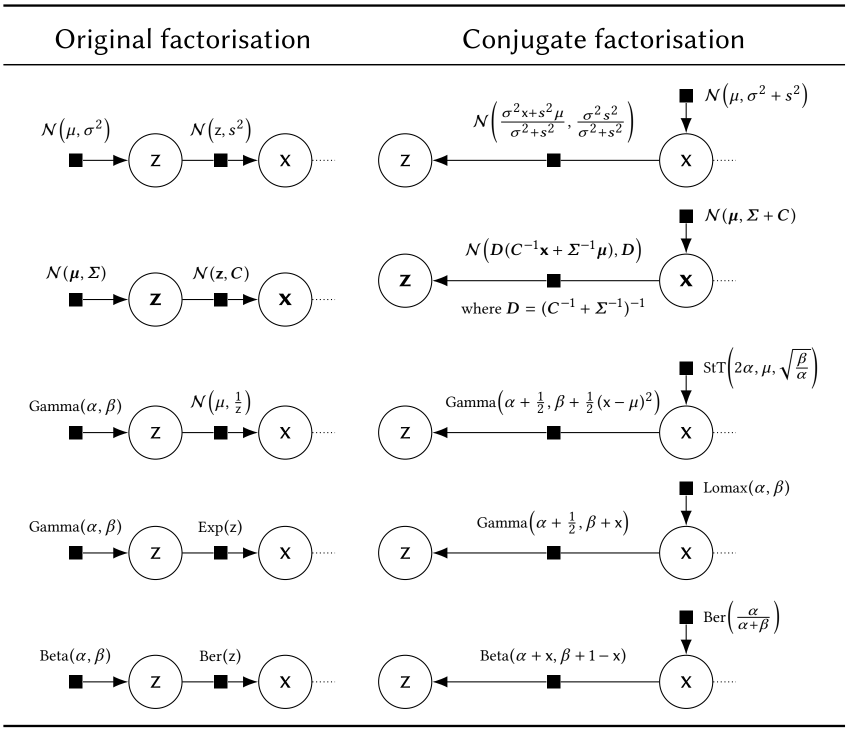

Original factorisation
Conjugate factorisation
$\mathcal{N}(\mu, \sigma^2)$
$\mathcal{N}(z, s^2)$
z
x
$\mathcal{N}\left(\frac{\sigma^2 x + s^2 \mu}{\sigma^2 + s^2}, \frac{\sigma^2 s^2}{\sigma^2 + s^2}\right)$
$\mathcal{N}(\mu, \sigma^2 + s^2)$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\mathcal{N}(\mathbf{z}, \boldsymbol{C})$
$\mathcal{N}(\boldsymbol{D}(\boldsymbol{C}^{-1}\mathbf{x} + \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}), \boldsymbol{D})$
where $\boldsymbol{D} = (\boldsymbol{C}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma} + \boldsymbol{C})$
Gamma(α, β)
$\mathcal{N}(\mu, \frac{1}{z})$
Gamma$\left(\alpha + \frac{1}{2}, \beta + \frac{1}{2}(x - \mu)^2\right)$
StT$\left(2\alpha, \mu, \sqrt{\frac{\beta}{\alpha}}\right)$
Gamma(α, β)
Exp(z)
Gamma$\left(\alpha + \frac{1}{2}, \beta + x\right)$
Lomax(α, β)
Beta(α, β)
Ber(z)
Beta$(\alpha + x, \beta + 1 - x)$
Ber$\left(\frac{\alpha}{\alpha + \beta}\right)$

Table 1.4: Factor graph illustrations of conjugate distributions.

1.3.5 Conjugacy and exact inference

Some densities have a *conjugacy* property that can simplify inference. If x and z are two random variables in a model then the joint density on the two variable can be factorised as

$$\mathrm{p}_{\mathrm{x,z}}(x, z) = \mathrm{p}_{\mathrm{x|z}}(x \mid z)\mathrm{p}_{\mathrm{z}}(z) = \mathrm{p}_{\mathrm{z|x}}(z \mid x)\mathrm{p}_{\mathrm{x}}(x). \tag{1.71}$$

For certain pairs of $\mathrm{p}_{\mathrm{x|z}}$ and p_{z} the corresponding $\mathrm{p}_{\mathrm{z|x}}$ and p_{x} have closed-form expressions. Some examples are shown in Table 1.4. In all of these examples $\mathrm{p}_{\mathrm{x|z}}$ is a density for an *exponential family distribution*. For every exponential family distribution density $\mathrm{p}_{\mathrm{x|z}}$ there exists a p_{z} such that $\mathrm{p}_{\mathrm{z|x}}$ is of the same parametric family as p_{z}.

A conditional density $\mathrm{p}_{\mathbf{u}|\mathbf{v}}$ *is from the exponential family if it can be written as* $\mathrm{p}_{\mathbf{u}|\mathbf{v}}(\boldsymbol{u} \mid \boldsymbol{v}) = \frac{h(\boldsymbol{u}) \exp(\boldsymbol{\eta}(\boldsymbol{v})^{\mathsf{T}} \boldsymbol{t}(\boldsymbol{u}))}{z(\boldsymbol{v})}$, *with* $\boldsymbol{\eta}(\boldsymbol{v})$ *termed the* natural parameters *and* $\boldsymbol{t}(\boldsymbol{u})$ *termed the* sufficient statistics.

If x corresponds to an observed variable in the model, then if evaluating conditional expectations $\mathbb{E}[f(\mathrm{z}) \mid \mathrm{x}]$ the conjugacy property means that conditional density $\mathrm{p}_{\mathrm{z|x}}$ which f should be integrated against has a closed form expression. Often for simple f, e.g. $f(z) = z$ or $f(z) = z^2$, the conditional expectations will have closed form solutions in such cases (i.e. corresponding to moments of the distribution defined by $\mathrm{p}_{\mathrm{z|x}}$). Even when f is more complex, typically generating independent random samples from $\mathrm{p}_{\mathrm{z|x}}$ in such cases will be possible, simplifying

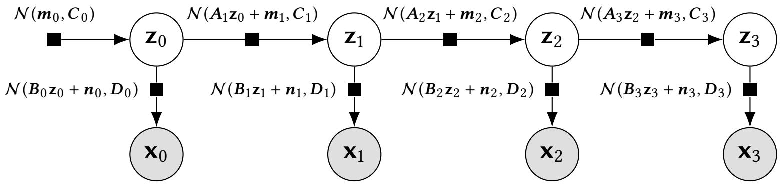

$\mathcal{N}(\boldsymbol{m}_0, C_0)$
$\mathcal{N}(A_1\mathbf{z}_0 + \boldsymbol{m}_1, C_1)$
$\mathcal{N}(A_2\mathbf{z}_1 + \boldsymbol{m}_2, C_2)$
$\mathcal{N}(A_3\mathbf{z}_2 + \boldsymbol{m}_3, C_3)$
$\mathbf{z}_0$
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_3$
$\mathcal{N}(B_0\mathbf{z}_0 + \boldsymbol{n}_0, D_0)$
$\mathcal{N}(B_1\mathbf{z}_1 + \boldsymbol{n}_1, D_1)$
$\mathcal{N}(B_2\mathbf{z}_2 + \boldsymbol{n}_2, D_2)$
$\mathcal{N}(B_3\mathbf{z}_3 + \boldsymbol{n}_3, D_3)$
$\mathbf{x}_0$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$

Figure 1.11: Factor graph for a latent linear dynamical system model.

the use of Monte Carlo methods (which will be introduced in the next chapter). If both z and x are latent variables then the conjugacy property is also useful as in this case z can be analytically marginalised out of conditional expectations of functions which do not depend on z.

A Markov process is a stochastic process such that future states are conditionally independent of past states given the current state.

Various algorithms have been developed to exploit conjugacy in restricted classes of probabilistic models to perform efficient exact inference. Inference in *latent linear dynamical systems*, an example of which is shown as a factor graph in Figure 1.11, can be efficiently performed with the recursive Kalman filtering and smoothing algorithms [54]. Closely related are the forward and backward updates for inference in *hidden Markov models* [113] which have the same factorisation structure as latent linear dynamical systems but use a discrete rather than real-valued latent state. The *junction tree algorithm* [63] is a general purpose algorithm for performing exact inference in undirected graphical models of discrete random variables which exploits conditional independency structure to efficiently decompose the inference into local computations; the computational cost of the algorithm scales exponentially with the number of nodes in the largest clique in the model and so is mainly relevant for graphs with relatively small maximal cliques.

1.3.6 A note on Bayesian terminology

The system of inference that we have described would typically be identified as being *Bayesian*. This name arises because of the central importance of *Bayes' theorem* (1.7) in defining the conditional probability of unobserved variables given observations.

While often the inference problems we will discuss in this thesis will be motivated from a Bayesian standpoint, the use of probabilistic modelling and resulting need to deal with the computational challenges of computing integrals in high dimensional spaces are not unique to Bayesian statistics. For instance many of the approximate inference methods we will discuss in the next two chapters were originally developed for use in studying statistical physics problems such as the phase transitions of the *Ising spin model* (which we earlier encounted under the alternative name of the Boltzmann machine model). As our main interest in this thesis will be the computational aspects of inference rather than specific applications, we will therefore generally prefer to refer to probabilistic modelling and inference in general terms rather than Bayesian inference in particular.

For completeness we will briefly review the nomenclature typically used in Bayesian inference. For a model with observed variables $\mathbf{x} \in X$ and unobserved variables $\mathbf{z} \in Z$, often referred to as *model parameters* in a Bayesian setting, the standard presentation of Bayesian inference assumes that the joint probability is specified by a density $\mathrm{p}_{\mathbf{x},\mathbf{z}} = \frac{\mathrm{d}\mathrm{P}_{\mathbf{x},\mathbf{z}}}{\mathrm{d}(\mu_{\mathbf{x}} \times \mu_{\mathbf{z}})}$ which naturally factorises as $\mathrm{p}_{\mathbf{x},\mathbf{z}}(\boldsymbol{x}, z) = \mathrm{p}_{\mathbf{x}|\mathbf{z}}(\boldsymbol{x} \,|\, z)\, \mathrm{p}_{\mathbf{z}}(z)$.

In this case the probability distribution defined by the marginal density on the model parameters $\mathrm{p}_{\mathbf{z}}$ is typically termed the *prior distribution*, with the intepretation that it captures our beliefs about the parameters before observing data. The conditional density on the observed variables given parameters $\mathrm{p}_{\mathbf{x}|\mathbf{z}}$ is often referred to as the *likelihood* or sometimes the *statistical model* [97].

The distribution defined by the conditional density on the parameters given the observations $\mathrm{p}_{\mathbf{z}|\mathbf{x}}$ is termed the *posterior distribution*, with this naming indicative of it representing our beliefs about the parameters after observing data. The posterior density can be calculated from the prior and likelihood using the definition for conditional density given in (1.44). Analagously to the definition for probabilities in (1.7) the resulting relationship is often also termed Bayes' theorem

$$\mathrm{p}_{\mathbf{z}|\mathbf{x}}(z \,|\, \boldsymbol{x}) = \frac{\mathrm{p}_{\mathbf{x}|\mathbf{z}}(\boldsymbol{x} \,|\, z)\, \mathrm{p}_{\mathbf{z}}(z)}{\mathrm{p}_{\mathbf{x}}(\boldsymbol{x})} \qquad \forall z \in Z,\ \boldsymbol{x} \in X : \mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) > 0. \tag{1.72}$$

Inference is then typically posed as the problem of computing the posterior distribution. This is typically not available in a closed form as evaluating the denominator in (1.72), $\mathrm{p}_{\mathbf{x}}$, sometimes termed the *model evidence*, requires integrating the joint density over the parameters

$$\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \int_Z \mathrm{p}_{\mathbf{x},\mathbf{z}}(\boldsymbol{x}, z)\, \mathrm{d}z = \int_Z \mathrm{p}_{\mathbf{x}|\mathbf{z}}(\boldsymbol{x} \,|\, z)\, \mathrm{p}_{\mathbf{z}}(z)\, \mathrm{d}z. \tag{1.73}$$

Other than in special cases such as when $\mathrm{p}_{\mathbf{z}}$ and $\mathrm{p}_{\mathbf{x}|\mathbf{z}}$ are both in the exponential family and form a conjugate pair such as those shown in

Table 1.4, this integral does not have a closed form solution and typically Z will be of a dimensionality which means quadrature will be too expensive. As $\mathsf{p}_{\mathsf{x}}(\boldsymbol{x})$ is found by marginalising out $\mathbf{z}$ from the product of the prior and likelihood, it is sometimes termed the *marginal likelihood*, though this name obscures that it is also dependent on the prior.

The usage of the term likelihood in Bayesian inference is overloaded: while it is often informally used to refer to the conditional density in $\mathsf{p}_{\mathbf{x}|\mathbf{z}}$ in Bayes' theorem (which is often summarised as *posterior is proportional to likelihood times prior*), the likelihood is usually formally defined as a function of the parameters (given fixed values for the observed variables) with notation such as $\ell(\boldsymbol{z} \mid \boldsymbol{x}) = \mathsf{p}_{\mathbf{x}|\mathbf{z}}(\boldsymbol{x} \mid \boldsymbol{z})$ sometimes used to emphasise this. This usage is particularly common when discussing *maximum likelihood* methods which find values of the parameters which maximise the likelihood (given data). In this latter interpretation it makes sense to refer to the likelihood of the parameters, but not the likelihood of observed variables (or observations / data). This leads to recommendations to refer to 'the likelihood of the parameters given the [observed] data' [66], which given the construct being discussed is actually a density on the observed data given parameter values is, in our opinion, not particularly clear.

In this thesis we will avoid using the terms *likelihood* and *marginal likelihood*. We will generally prefer to refer to individual conditional and marginal densities (or probabilities) explicitly using the notation introduced earlier in this chapter, though we will use the terms prior, posterior and model evidence as qualifiers where appropriate.

We will also prefer to consider a probabilistic model as defining a joint probability measure on observed and unobserved variables $\mathsf{P}_{\mathbf{x},\mathbf{z}}$, without necessarily making further assumptions of how this joint probability is specified. Commonly the model will be defined via an explicit joint density $\mathsf{p}_{\mathbf{x},\mathbf{z}}$ given for example by a directed factor graph, but for example in the case of simulator models, the model may instead by defined procedurally by code for sampling from the joint probability, with the joint density only implictly defined.

A further common special case is when the model is specified by a marginal density on the unobserved variables $\mathsf{p}_{\mathbf{z}}$ (a prior in Bayesian terms) and a conditional density on observed variables given unobserved variables $\mathsf{p}_{\mathbf{x}|\mathbf{z}}$ (a 'likelihood'), this factorisation naturally leading to the

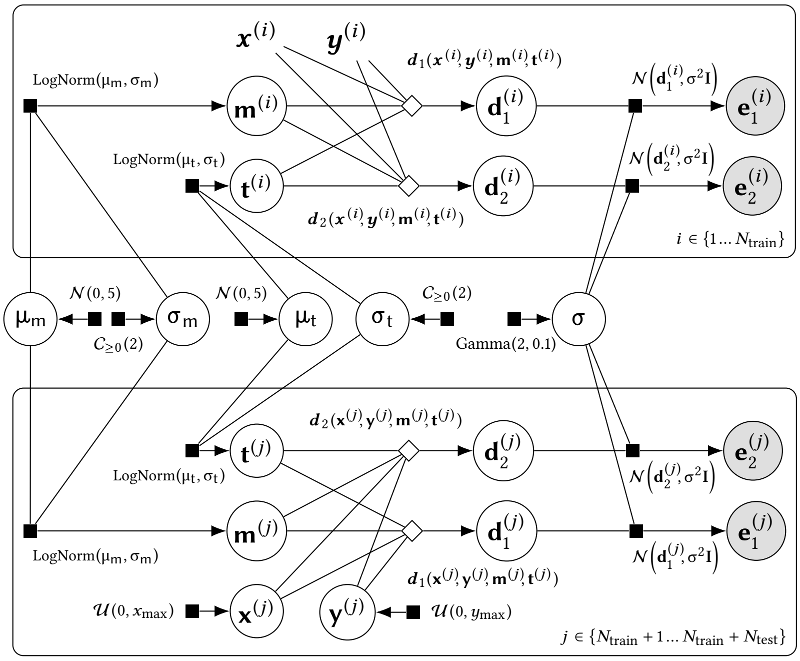

$\boldsymbol{x}^{(i)}$
$\boldsymbol{y}^{(i)}$
$d_1(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \mathbf{m}^{(i)}, \mathbf{t}^{(i)})$
$\mathrm{LogNorm}(\mu_m, \sigma_m)$
$\mathbf{m}^{(i)}$
$\mathbf{d}_1^{(i)}$
$\mathcal{N}\left(\mathbf{d}_1^{(i)}, \sigma^2\mathbf{I}\right)$
$\mathbf{e}_1^{(i)}$
$\mathrm{LogNorm}(\mu_t, \sigma_t)$
$\mathbf{t}^{(i)}$
$\mathbf{d}_2^{(i)}$
$\mathcal{N}\left(\mathbf{d}_2^{(i)}, \sigma^2\mathbf{I}\right)$
$\mathbf{e}_2^{(i)}$
$d_2(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}, \mathbf{m}^{(i)}, \mathbf{t}^{(i)})$
$i \in \{1 \dots N_{\mathrm{train}}\}$
$\mathcal{N}(0, 5)$
$\mathcal{N}(0, 5)$
$C_{\geq 0}(2)$
μ_m
σ_m
μ_t
σ_t
σ
$C_{\geq 0}(2)$
$\mathrm{Gamma}(2, 0.1)$
$d_2(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, \mathbf{m}^{(j)}, \mathbf{t}^{(j)})$
$\mathbf{t}^{(j)}$
$\mathbf{d}_2^{(j)}$
$\mathbf{e}_2^{(j)}$
$\mathrm{LogNorm}(\mu_t, \sigma_t)$
$\mathcal{N}\left(\mathbf{d}_2^{(j)}, \sigma^2\mathbf{I}\right)$
$\mathbf{m}^{(j)}$
$\mathbf{d}_1^{(j)}$
$\mathbf{e}_1^{(j)}$
$\mathrm{LogNorm}(\mu_m, \sigma_m)$
$\mathcal{N}\left(\mathbf{d}_1^{(j)}, \sigma^2\mathbf{I}\right)$
$d_1(\mathbf{x}^{(j)}, \mathbf{y}^{(j)}, \mathbf{m}^{(j)}, \mathbf{t}^{(j)})$
$\mathcal{U}(0, x_{\max})$
$\mathbf{x}^{(j)}$
$\mathbf{y}^{(j)}$
$\mathcal{U}(0, y_{\max})$
$j \in \{N_{\mathrm{train}} + 1 \dots N_{\mathrm{train}} + N_{\mathrm{test}}\}$

Figure 1.12: Observing Dark Worlds hierarchical model.

form given for Bayes' theorem in (1.72). However the assumption that the joint density can be easily factorised into closed form densities $\mathrm{p}_{\mathbf{z}}$ and $\mathrm{p}_{\mathbf{x}|\mathbf{z}}$ is not a requirement to evaluate the posterior density $\mathrm{p}_{\mathbf{z}|\mathbf{x}}$, and there are inference problems where this will not apply.

As a final comment on terminology, we will prefer to refer to unobserved or latent variables (which we will use interchangeably) rather than parameters when these quantities are being inferred as opposed to set to fixed values. Parameters and latent variables are sometimes used distinctly, in particular defining parameters as unobserved variables that are of a fixed number and which all observed variables are dependent on, and latent variables as unobserved variables which only the observed variables associated with a particular data point are dependent upon and which increase in number with the quantity of observed datapoints [12]. When this distinction is helpful we will instead follow the terminology used in [14] and refer to the former as *global latent variables* and the latter as *local latent variables*.

1.3.7 Hierarchical modelling

When introducing the proposed model (Figure 1.9) for the *Observing Dark Worlds* problem we glossed over how the parameters σ, μ_m, σ_m, μ_t and σ_t were chosen, assuming for simplicity they had somehow been set to reasonable values. Rather than considering these quantities as fixed parameters, a more Bayesian approach would be instead to also treat them as further unobserved variables to infer.

In particular if we now consider σ, μ_m, σ_m, μ_t and σ_t as (global) latent variables, we can encode our prior beliefs about what are plausible values for the variables by setting densities on each. For example, for the intrinsic ellipticity scale variable σ we might use $\mathrm{Gamma}(2, 0.1)$. This has support only for positive values as required for a scale variable, and reflects the known properties of the intrinsic ellipticities - that they have non-zero variability (with the density tending to zero as $\sigma \to 0$) and that they are bounded to unit magnitude and hence their standard deviation would be expected to be $\ll 1$. We might have less strong prior beliefs about the range of plausible values for the halo mass and core radii distribution, and so choose to use concordantly less constraining prior distributions on the scale and location variables for these distributions. For example a prior of $\mathcal{N}(0, 5)$ on the location variables μ_m and μ_t and a prior of $\mathcal{C}_{\geq 0}(2)$ on the scale variables σ_m and σ_t, supports a wide range of values for these variables as being reasonable while still making a weak assumption that extreme values are implausible.

In the *Observing Dark Worlds* competition, two distinct sets of data were provided — a training set of $N_\mathrm{train} = 300$ of cluster images in which the observed galaxy ellipticities, galaxy coordinates and halo positions were all provided, and a test set of $N_\mathrm{test} = 120$ images where only the observed galaxy ellipticities and galaxy coordinates are provided. Assuming both the training and test data are iid we can form a *hierarchical model* for the problem which specifies a joint density across the observed and local latent variables for all of the training and test set clusters as well as the global latent variables σ, μ_m, σ_m, μ_t and σ_t. A factor graph for the proposed hierarchical model is shown in Figure 1.12. Compared to the factor graph in Figure 1.9 for a single cluster image some detail in the model structure for each cluster has been hidden to make the higher level structure clearer, and the galaxy positions, which are all observed, have been omitted. The vector notation introduced when discussing inference in the model has also been used to remove the need for plates indexed across galaxies and halos.

When making predictions using the hierarchical model, the local latent variables $\{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \mathbf{m}^{(i)}, \mathbf{t}^{(i)}\}$ are now jointly dependent when not conditioning on the values of the global latent variables σ, μ_m, σ_m, μ_t and σ_t.

It is therefore necessary to jointly integrate across all of the local latent variables associated with each training and test set cluster when evaluating conditional expectations given the observed data. The resulting integral is over a space of several thousand dimensions. Given the infeasibility of using quadrature to compute conditional expectations of the halo positions for even a single cluster image, clearly in this case the need for computational methods for inference which scale to high dimensionalities is even more stark.

1.3.8 Model comparison

So far we have discussed making inferences about the unobserved variables in a single fixed model. An important second level of inference is comparing between competing models for the same observed data. As we will see this can be treated consistently with a simple extension to the existing probabilistic framework we have discussed.

As a simple motivating example, we can consider comparing our proposed model for the *Observing Dark Worlds* problem, in which we assumed that the magnitude of the shear distortion had the functional dependence given in (1.54), to an alternative model which assumes

$$f_h(u,v) = \frac{\mathsf{m}_h}{r_h} \quad \text{where} \quad r_h = \sqrt{(\mathsf{x}_h - u)^2 + (\mathsf{y}_h - v)^2}. \qquad (1.74)$$

This model is simpler in the sense of requiring only one additional latent variable, m_h, per halo (compared to both m_h and t_h), but produces a non-realistic infinite magnitude distortion at zero radial distances. Given the observed data, we would ideally like to be able to make a judgement as to which of the two proposed models better describes the data. To be useful this comparison must take into account the relative complexity of the models; a model with more free variables will generally be able to fit observed data more closely, however *Ockham's Razor* (and corresponding empirical evidence of the loss of predictive power of overly complex models) suggests we should prefer simpler models where possible. By marginalising over the unobserved variables in a model, the probabilistic model comparison framework we will describe automatically embodies Ockham's Razor [66].

Ockham's Razor is a philoshopical principle, commonly attributed to the 14th century Franciscan friar William of Ockham, that states if there exist multiple explanations for observations, all else being equal we should prefer the simplest.

The general model comparison set up we will assume is that we have a finite set of M models, indexed by an *indicator* variable $\mathsf{m} \in \{1 \dots M\}$. All models share the same observed variables $\mathbf{x}$, and there are a set

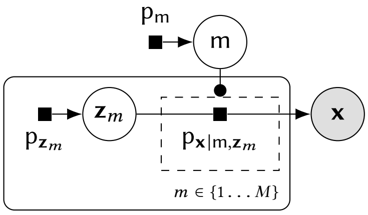

p_{m}
m
$\mathbf{z}_m$
$\mathsf{p}_{\mathbf{z}_m}$
$\mathsf{p}_{\mathbf{x}|\mathsf{m},\mathbf{z}_m}$
x
$m \in \{1 \ldots M\}$

Figure 1.13: Factor graph for inference over multiple models.

of per model vectors of unobserved variables $\{\mathbf{z}_m\}_{m=1}^{M}$ which are assumed to be independent (before conditioning on observations). More complex structures could be assumed such as the models sharing a set of common unobserved variables, however we only consider the case of independent models here. The joint density on the observations, model indicator and latent variables is then assumed to factorise as

$$\mathsf{p}_{\mathbf{x},\mathsf{m},\mathbf{z}_1,\ldots,\mathbf{z}_M}(\boldsymbol{x}, m, z_1, \ldots, z_M) = \\ \mathsf{p}_{\mathbf{x}|\mathsf{m},\mathbf{z}_m}(\boldsymbol{x} \mid m, z_m)\, \mathsf{p}_{\mathsf{m}}(m) \prod_{n=1}^{M} \mathsf{p}_{\mathbf{z}_n}(z_n). \tag{1.75}$$

The marginal density on the model indicator p_{m} represents our prior beliefs about the relative probabilities of the models before observing data. Importantly the value of the model indicator variable m selects the relevant per model conditional density on the observed variables given latent variables $\mathsf{p}_{\mathbf{x}|\mathsf{m},\mathbf{z}_m}$; this represents the assumption that conditioned on the model indicator assuming a particular model index m the observed variables are conditionally independent of the latent variables of all other models $\mathbf{x} \perp \{\mathbf{z}_n\}_{n \neq m} \mid \mathsf{m} = m$. An extension to factor graphs, *gates* [76], can be used to represent such context-dependent conditional independencies. Figure 1.13 shows a gated factor graph of equation 1.75, with the gate indicated by the dashed box.

Given this computational set up, the task in model comparison is then to compute the relative probabilities of each of the models given observed data. These probabilities[6] are given by

$$\mathsf{p}_{\mathsf{m}|\mathbf{x}}(m \mid \boldsymbol{x}) = \frac{\mathsf{p}_{\mathbf{x}|\mathsf{m}}(\boldsymbol{x} \mid m)\, \mathsf{p}_{\mathsf{m}}(m)}{\sum_{n=1}^{M} (\mathsf{p}_{\mathbf{x}|\mathsf{m}}(\boldsymbol{x} \mid n)\, \mathsf{p}_{\mathsf{m}}(n))}, \tag{1.76}$$

which can be seen as a direct analogue to Bayes' theorem for the posterior density on unobserved random variables for a single model. The

[6] As m is a discrete random variable the probability of the event of m taking a value in the singleton set $\{m\}$ given that $\mathbf{x} = \boldsymbol{x}$ is equal to the density $\mathsf{p}_{\mathsf{m}|\mathbf{x}}(m \mid \boldsymbol{x})$.

key quantities needed to evaluate the model posterior probabilities are the marginal densities $\mathrm{p}_{\mathbf{x}|\mathrm{m}}(\boldsymbol{x} \mid m)$ evaluated at the observed data. Computing these values requires marginalising out the latent variables from the per model joint densities on observed and latent variables

$$\mathrm{p}_{\mathbf{x}|\mathrm{m}}(\boldsymbol{x} \mid m) = \int_{Z_m} \mathrm{p}_{\mathbf{x}|\mathrm{m},\mathbf{z}_m}(\boldsymbol{x} \mid m, z)\,\mathrm{p}_{\mathbf{z}_m}(z)\,\mathrm{d}z. \qquad (1.77)$$

This value is equal to the denominator in Bayes' theorem (1.72), this explaining the naming of this term as the *model evidence*.

As with evaluating conditional expectations of functions of the latent variables in a model given observations, evaluating the model evidence values requires integrating across the space of all latent variables. In most cases this integral will not have an analytic solution and the number of latent variables will be sufficiently large to make numerical quadrature methods impractical. The key computational challenge in being able to perform probabilistic model comparison with complex high dimensional models is therefore again being able to efficiently to compute integrals in high dimensional spaces.

1.4 SUMMARY

In this chapter we reviewed the probabilistic modelling framework that will form the basis for the methods we will introduce in the rest of the thesis. We began by introducing some of the underlying concepts from probability theory, in particular focusing on the manipulation random variables which are a key abstraction for explaining much of theory behind the methods in this thesis. We illustrated the use of factor graphs to efficiently communicate the structure of complex probabilistic models and discussed their close links to computation graphs which allow efficient calculation of the derivatives of the outputs of complex functions of a large numbers of variables with respect to their inputs. We concluded by motivating the computational challenges of performing inference in complex probabilistic models involving large numbers of unobserved variables. In particular we identified the key computational requirement as being able to evaluate integrals across high-dimensional spaces. In the next chapter we will introduce some of the computational methods which have been proposed to address the challenges of finding solutions to inference problems.

2 APPROXIMATE INFERENCE

In the previous chapter we motivated the claim that the key computational challenge in performing inference in probabilistic models is being able to evaluate integrals with respect to probability measures defined on high-dimensional spaces. Although in restricted cases such as conjugate exponential family models, the integrals involved in inference can be solved analytically, to exploit the full flexibility of probabilistic modelling we need to be able to compute such integrals in more general cases. As discussed in the previous chapter, for models with even moderate numbers of latent variables, numerical quadrature approaches to evaluating integrals are computationally infeasible due to the exponential scaling of computation cost with dimension.

Truth is much too complicated to allow anything but approximations.
—John von Neumann

In this chapter we will review some of the key algorithms proposed for computing *approximate* solutions to inference problems. A unifying aspect to all of these methods is trading off some loss of the accuracy of the answers provided to inferential queries, for a potentially significant increase in computational tractability. The literature on *approximate inference* methods is vast and so necessarily this chapter will only form a partial review of the available methods. We will therefore concentrate on those approaches which are directly relevant to this thesis.

Approximate inference methods can be coarsely divided in to two groups: those in which a more tractable approximation to the target probability measure of interest is found by optimising the approximation to be ‘close’ in some sense to the target measure; methods in which the integrals with respect to the target measure are estimated by computing weighted sums over points sampled from a probability measure over the target state space. As with any such binary classification of such a broad and diverse topic there are however methods which combine aspects of both these approaches. The chapter will therefore be begin with sections reviewing the key sampling and optimisation based approaches to approximate inference, before concluding with a discussion of some of the methods proposed which use a hybrid approach.

2.1 SAMPLING APPROACHES

A key observation in the previous chapter was that inference at both the level of computing conditional expectations of latent variables in a model and in evaluating evidence terms to allow model comparison, will for most models of interest correspond to being able to integrate (potentially constant) functions against a probability density defined with respect to a base measure[1]. In particular we wish to be able to compute integrals of the form

$$\int_X f(\boldsymbol{x})\,\mathrm{d}P(\boldsymbol{x}) = \int_X f(\boldsymbol{x})\,p(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) \tag{2.1}$$

where p is the density of a target distribution P on a space X with respect to a base measure μ and f is a measurable function. For instance in the case of computing the *posterior mean* in a Bayesian inference problem with observed variables $\mathbf{y}$ and latent variables $\mathbf{x}$ where the posterior density $\mathsf{p}_{\mathbf{x}|\mathbf{y}}$ is defined with respect to the D-dimensional Lebesgue measure, we would have $p(\boldsymbol{x}) = \mathsf{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x} \,|\, \boldsymbol{y})$ for an observed $\boldsymbol{y}$, $\mu(\boldsymbol{x}) = \lambda^D(\boldsymbol{x})$ and $f(\boldsymbol{x}) = \boldsymbol{x}$. Often we will only be able to evaluate p up to an unknown unnormalising constant i.e. $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$ with we able to evaluate $\tilde{p}$ pointwise but Z intractable to compute. For example in a Bayesian inference setting $\tilde{p}(\boldsymbol{x})$ would be the joint density $\mathsf{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y})$ and Z the model evidence $\mathsf{p}_{\mathbf{y}}(\boldsymbol{y})$. When peforming inference in undirected graphical models, we would instead have that $\tilde{p}$ is the product of clique potentials and Z the corresponding normaliser.

The key idea of the methods we will discuss in this section is that we can estimate (2.1) by generating a set of random samples from a probability distribution defined on X and then computing a (potentially weighted) average of the value of the function f evaluated at these sample points. The most obvious approach is to sample independently from the probability distribution defined by the target density p. As we will see this is not necessarily feasible to do for the complex target densities defined on high dimensional spaces, however a host of related methods for generating and using random samples to approximate integrals with respect to target densities arising from complex probabilistic models have been developed.

[1] There are models for which the corresponding probability measure is not absolutely continuous with respect to another measure and so cannot be represented by a density, however we will concentrate for now on the common case were a density exists.

2.1.1 Monte Carlo method

The framework that unifies all of the methods we will discuss in this section is the *Monte Carlo method* for integration [121]. We will briefly describe the key ideas and properties of Monte Carlo integration.

The eponym of the Monte Carlo method is a Monocan casino, favoured haunt of the uncle of Stanisław Ulam, one of the method's inventors.

Let $\mathbf{x}$ be a random (vector) variable distributed according to the target density i.e. $\mathrm{p}_{\mathbf{x}} = \frac{\mathrm{dP}_{\mathbf{x}}}{\mathrm{d}\mu} = p$. Given an arbitrary measurable function $f : X \to \mathbb{R}$ we define a random variable $\mathrm{f} = f(\mathbf{x})$. Our task is to compute expectations $\mathbb{E}[\mathrm{f}] = \bar{f}$ corresponding to the integral (2.1). We assume that $\mathbb{E}[\mathrm{f}]$ exists and both $\mathbb{E}[\mathrm{f}]$ and $\mathbb{V}[\mathrm{f}]$ are finite.

For now we assume we have a way of generating values of N random variables $\{\mathbf{x}_n\}_{n=1}^N$, each marginally distributed according to the target density i.e. $\mathrm{p}_{\mathbf{x}_n} = p \ \forall n \in \{1 \dots N\}$. We will initially not require any further properties on the joint distribution across all N variables. We define random variables $\{\mathrm{f}_n\}_{n=1}^N$ and $\hat{\mathrm{f}}_N$ by

$$\mathrm{f}_n = f(\mathbf{x}_n) \quad \forall n \in \{1 \dots N\} \quad \text{and} \quad \hat{\mathrm{f}}_N = \frac{1}{N} \sum_{n=1}^{N} \mathrm{f}_n. \tag{2.2}$$

Due to linearity of the expectation operator, we have that

$$\mathbb{E}\left[\hat{\mathrm{f}}_N\right] = \frac{1}{N} \sum_{n=1}^{N} \mathbb{E}[\mathrm{f}_n] = \frac{1}{N} \sum_{n=1}^{N} \bar{f} = \bar{f} \tag{2.3}$$

and so that in expectation $\hat{\mathrm{f}}_N$ is equal to $\bar{f}$, i.e. realisations of $\hat{\mathrm{f}}_N$ are unbiased estimators of $\bar{f}$. Note that this result does not require any independence assumptions about the generated random variables. Now considering the variance of $\hat{\mathrm{f}}_N$ it can be shown that

$$\mathbb{V}\left[\hat{\mathrm{f}}_N\right] = \frac{\mathbb{V}[\mathrm{f}]}{N} \left(1 + \frac{2}{N} \sum_{n=1}^{N-1} \sum_{m=1}^{n-1} \frac{\mathbb{C}[\mathrm{f}_n, \mathrm{f}_m]}{\mathbb{V}[\mathrm{f}]} \right). \tag{2.4}$$

If the generated random variables $\{\mathbf{x}_n\}$ and so $\{\mathrm{f}_n\}$ are independent, then $\mathbb{C}[\mathrm{f}_n, \mathrm{f}_m] = 0 \ \forall m \neq n$. In this case (2.4) reduces to

$$\mathbb{V}\left[\hat{\mathrm{f}}_N\right] = \frac{\mathbb{V}[\mathrm{f}]}{N}, \tag{2.5}$$

i.e. the variance of the *Monte Carlo estimate* $\hat{\mathrm{f}}_N$ for $\bar{f}$ is inversely proportional to the number of generated random samples N. Importantly this scaling does not depend on the dimension of $\mathbf{x}$.

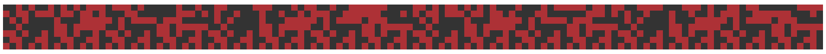


Figure 2.1: Binary representation of linear congruential generator sequence $s_{n+1} = 37 s_n + 61 \bmod 128$. Columns left to right represents successive integer states in sequence. From least significant (bottom) to most significant (top), the bits in each column have patterns repeating with periods 2, 4, 8, 16, 32, 64, 128.

Therefore if we can generate a set of independent random variables from the target density, we can estimate expectations that asymptotically tend to the true value as N increases, with a typical deviation from the true value (as measured by the standard deviation, i.e. the square root of variance) that is $\mathcal{O}\left(N^{-\frac{1}{2}}\right)$. In comparison a fourth-order quadrature method such as *Simpson's rule* has an error that is $\mathcal{O}\left(N^{-\frac{4}{D}}\right)$ for a grid of N points uniformly spaced across a D dimensional space. Asymptotically for $D > 8$, Monte Carlo integration will therefore give better convergence than Simpson's rule, and even for smaller dimensions the large constant factors in the error dependence can sometimes favour Monte Carlo.

Note that computing Monte Carlo estimates from independent random variables is not optimal in terms of minimising $\mathbb{V}\left[\hat{\mathrm{f}}_N\right]$ for a given f; the covariance terms in (2.4) can be negative which can reduce the overall variance. A wide range of *variance reduction* methods have been proposed to exploit this and produce lower variance of Monte Carlo estimates for a given f [61]. Although these methods can be very important in pratice for achieving an estimator with a practical variance for a specific f of interest, we will generally concentrate on the case where we do not necessarily know f in advance and so cannot easily exploit these methods.

2.1.2 Pseudo-random number generation

The generation of random numbers is too important to be left to chance.
—Robert R. Coveyou

Virtually all statistical computations involving random numbers in practice make use of *pseudo-random number generators* (PRNGs). Rather than generating samples from truly random processes[2], PRNGs produce deterministic[3] sequences of integers in a fixed range that nonetheless maintain many of the properties of a random sequence.

[2] In a true random process it is impossible to precisely predict the next value in the sequence given the previous values.

[3] The sequences are determnistic in the sense that if the generator internal state is known all values in the sequence can be reconstructed exactly.

In particular through careful choice of the updates, sequences with a very long period (number of iterations before the sequence begins repeating), a uniform distribution across the numbers in the sequence range and low correlation between successive states can be constructed. A very simple example of a class of PRNGs is the *linear congruential generator* [64] which obeys the recurrent update

$$s_{n+1} = (a s_n + c) \mod m \quad \text{with} \quad 0 < a < m,\ 0 \leq c < m, \tag{2.6}$$

with a, c and m integer parameters. If a, c and m are chosen appropriately, iterating the update (2.6) from an initial seed $0 \leq s_0 < m$, will produce a sequence of states which visits all the integers in $[0, m)$ before repeating. An example state sequence with $m = 128$ is shown in Figure 2.1. In practice, linear congruential generators produce sequences with poor statistical properties, particularly when used to generate random points in high dimensional spaces [67], hence most modern numerical computing libraries use more robust variants such as the *Mersenne-Twister* [71], which is used in all experiments in this thesis.

The raw output of a PRNG is an integer sequence, with typically the sequence elements uniformly distributed over all integers in a range $[0, 2^n)$ for some $n \in \mathbb{N}$. All real values are represented at a finite precision on computers, typically using a floating point representation [2] of *single* (24-bit mantissa) or *double* (53-bit mantissa) precision. Through an approriate linear transformation, the integer outputs of a PRNG can be converted to floating-point values uniformly distributed across a finite interval. PRNG implementations typically provide a primitive to generate floating-point values uniformly distributed on $[0, 1)$.

Given the ability to generate sequences of (effectively) independent samples from a uniform distribution $\mathcal{U}(0, 1)$, the question is then how to use these values to produce random samples from arbitary densities. This will be the subject of the following sub-sections.

2.1.3 Transform sampling

Samples from a large class of distributions can be generated by directly exploiting the transform of random variables relationships discussed in 1.1.4. In particular if $\mathbf{u}$ is D-dimensional vector of independent random variables with $\mathcal{U}(0, 1)$ marginal densities, then if $g : [0, 1)^D \to X$ is a

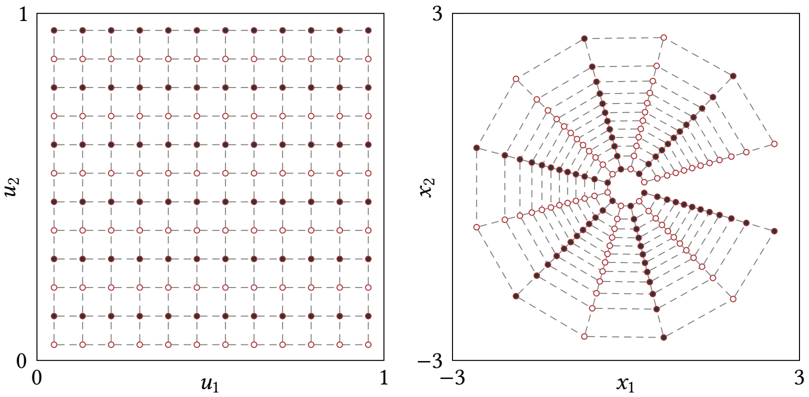


Figure 2.2: Visualisation of Box–Muller transform. Left axis shows uniform grid on $U = [0,1]^2$ and right-axis shows grid points after mapping through $\boldsymbol{g}$ in transformed space $X = \mathbb{R}^2$.

bijective map to a vector space $X \subseteq \mathbb{R}^D$, then if we define $\mathbf{x} = \boldsymbol{g}(\mathbf{u})$ and $\boldsymbol{h} = \boldsymbol{g}^{-1}$, then by applying (1.28) we have that

$$\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \left|\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|. \tag{2.7}$$

For example for $D = 2$, $X = \mathbb{R}^2$ and a bijective map $\boldsymbol{g}$ defined by

$$\begin{aligned} \boldsymbol{g}\left(\begin{array}{c} u_1 \\ u_2 \end{array}\right) &= \left[\begin{array}{c} \sqrt{-2\log u_1}\cos(2\pi u_1) \\ \sqrt{-2\log u_1}\sin(2\pi u_2) \end{array}\right], \\ \boldsymbol{h}\left(\begin{array}{c} x_1 \\ x_2 \end{array}\right) &= \left[\begin{array}{c} \exp\left(-\frac{1}{2}(x_1^2 + x_2^2)\right) \\ \frac{1}{2\pi}\arctan\left(\frac{x_1}{x_2}\right) \end{array}\right], \end{aligned} \tag{2.8}$$

then we have that the density of the transformed $\mathbf{x} = \boldsymbol{g}(\mathbf{u})$ is

$$\mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_1^2}{2}\right)\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x_2^2}{2}\right), \tag{2.9}$$

i.e. x_1 and x_2 are independent random variables with standard normal densities $\mathcal{N}(0,1)$. This is the *Box–Muller transform* [16], and allows generation of independent standard normal variables given a PRNG primitive for sampling from $\mathcal{U}(0,1)$. A visualisation of the transformation of space applied by the method is shown in Figure 2.2.

Due to the relatively high cost of the trigonometric function evaluations, more efficient alternatives to Box–Muller are usually used in practice to generate normal random variables such as a rejection sampling variant [67] (rejection sampling will be discussed in the next sub-section) or the *Ziggurat algorithm* [68] (which also generalises to other symmetric univariate distributions).

A general method for sampling from univarite densities is to use an inverse *cumulative distribution function* (CDF) transform. For a probability density p on a scalar random variable, the corresponding CDF $r : \mathbb{R} \to [0, 1]$ is defined as

$$r(x) = \int_{-\infty}^{x} p(v) \,\mathrm{d}v \implies \frac{\partial r(x)}{\partial x} = p(x). \tag{2.10}$$

If u is a standard uniform random variable and $\mathsf{x} = r^{-1}(\mathsf{u})$ then

$$\mathrm{p}_{\mathsf{x}}(x) = \left| \frac{\partial r(x)}{\partial x} \right| = p(x). \tag{2.11}$$

To be able to use the inverse CDF transform method we need to be able to evaluate r^{-1}. For many univariate densities the CDF r itself does not have a closed form solution. For some densities such as the standard normal $\mathcal{N}(0, 1)$ even though the CDF does not have an analytic form in terms of elementary functions it is common for numerical computing libraries to provide numerical approximations to both r and r^{-1} which are accurate to within small multiples of machine precision. In densities where a standard library function for the CDF is not available, Chebyshev polynomial approximations to the density can be used to efficient compute an approximation to the CDF and an iterative solver used for the inversion [83].

Although the inverse CDF transform method gives a general recipe for sampling from univariate densities, it is not easy to generalise to multivariate densities and even for univariate densities, alternatives can be simpler to implement and in some cases more numerically stable.

2.1.4 Rejection sampling

An important class of generic sampling methods, particularly due their use as a building block in other algorithms, is rejection sampling [81]. Rejection sampling uses the observation that to sample from a probablity density $p : X \to [0, \infty]$ it is sufficient to uniformly sample from the volume under the graph of $(\boldsymbol{x}, p(\boldsymbol{x}))$.

The key requirement in rejection sampling is to identify a *proposal density q* which can be independently sampled from and that upper bounds

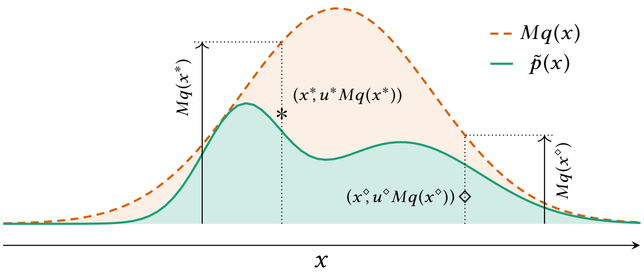


Figure 2.3: Visualisation of rejection sampling. The green curve shows the (unnormalised) target density $\tilde{p}$, the green region underneath representing the area we wish to sample points uniformly from. The dashed orange curve shows the scaled proposal density Mq, with the orange (plus green) region representing the area we uniformly propose values from. Two example proposals are shown: $\diamond$ is under the target density and so accepted; $*$ is outside of the green region and so would be rejected.

Algorithm 2 Rejection sampling.

Input: $\tilde{p}$: unnormalised target density, q : normalised proposal density, M : constant such that $\tilde{p}(\boldsymbol{x}) \leq Mq(\boldsymbol{x})\ \forall \boldsymbol{x} \in X$.
Output: Independent sample from target density $p(\boldsymbol{x}) = \frac{1}{Z}\tilde{p}(\boldsymbol{x})$.

```
repeat
    $\boldsymbol{x} \sim q$
    $u \sim \mathcal{U}(0, 1)$
until $uMq(\boldsymbol{x}) \leq \tilde{p}(\boldsymbol{x})$
return $\boldsymbol{x}$
```

the potentially unnormalised target density $\tilde{p}$ across its full support X when multiplied by a known constant M, i.e. $\tilde{p}(x) \leq Mq(x)\ \forall x \in X$. The first requirement to be able to generate independent samples from q can be met for example by distributions we can sample from using a transform sampling method as described in the previous subsection, e.g. standard normal. The second requirement is general more challenging, both from the perspective of ensuring the target density is upper bounded everywhere but also because as we will see how efficient the method is very dependent on how tight the bound is.

Algorithm 2 describes the rejection sampling method to produce a single independent sample from the target density. A visualisation of how the algorithm works for a univariate target density is shown in Figure 2.3. The overall aim is to generate points uniformly distributed across the green area under the (unnormalised) target density curve. This is achieved by generating points uniformly under the dashed orange curve corresponding to the scaled proposal density and then ac-

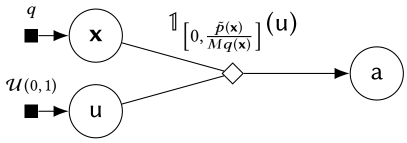

q
$\mathcal{U}(0,1)$
x
u
$\mathbb{1}_{\left[0,\frac{\tilde{p}(\mathbf{x})}{Mq(\mathbf{x})}\right]}(\mathrm{u})$
a

Figure 2.4: Factor graph of rejection sampling process.

cepting only those which are below the green curve. To generate a point under the dashed orange curve we first generate an x from the proposal density and then generate an auxilliary 'height' variable by sampling uniformly from $[0, Mq(x)]$. If the sampled height is below the green curve we accept (as in the $\diamond$ example in Figure 2.3) else we reject the sample (as in the $*$ example in Figure 2.3).

Figure 2.4 shows the rejection sampling generative process as a directed factor graph, with **x** be a random variable representing the proposal, u the uniform auxilliary variable drawn to sample the 'height' and a a binary variable that indicates whether the proposal is accepted (a = 1) or not (a = 0). By marginalising out u we have that that

$$\mathrm{p}_{\mathbf{x},\mathrm{a}}(\boldsymbol{x}, a) = q(\boldsymbol{x})\left(\frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}\right)^{a}\left(1 - \frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}\right)^{1-a}, \tag{2.12}$$

and further marginalising over the proposal **x**

$$\mathrm{p}_{\mathrm{a}}(a) = \left(\frac{Z}{M}\right)^{a}\left(1 - \frac{Z}{M}\right)^{1-a}. \tag{2.13}$$

Conditioning on the proposal being accepted we therefore have that

$$\mathrm{p}_{\mathbf{x}|\mathrm{a}}(\boldsymbol{x} \mid 1) = \frac{q(\boldsymbol{x})\frac{\tilde{p}(\boldsymbol{x})}{Mq(\boldsymbol{x})}}{\frac{Z}{M}} = \frac{\tilde{p}(\boldsymbol{x})}{Z} = p(\boldsymbol{x}). \tag{2.14}$$

Therefore the accepted proposals are distributed according to the target density as required. Further from (2.13) we have that the $\mathrm{p}_{\mathrm{a}}(1) = \frac{Z}{M}$. This suggests we can use the accept rate to estimate Z but also hints at the difficulty in finding a M which guarantees the upper bound requirement as for $\frac{Z}{M}$ to be a valid probability $M \geq Z$ i.e. M needs to be an upper bound on the unknown normalising constant Z. This relationship also suggests it is desirable to set M as small as possible to maximise the acceptance rate; for a fixed proposal density q this will involve setting M to a value such that $Mq(\boldsymbol{x}) = \tilde{p}(\boldsymbol{x})$ for at least one $\boldsymbol{x}$ (as for example in Figure 2.3).

Although rejection sampling can be an efficient method of sampling from univariate target densities (particularly in the case of log-concave densities where an adaptive variant is available [39]), it generally scales very poorly with the dimensionality of the target density. This is as the ratio of the volume under the target density to the volume under the scaled proposal (in terms of Figure 2.3 the ratio of the green area to the green plus orange regions), and so the probability of accepting a proposal, will tend become exponentially smaller as the dimensionality increases. This is the so-called *curse of dimensionality*. Therefore although rejection sampling can be a useful subroutine for generating random variables from low-dimensional densities, in general it is not a viable option for generating samples directly for high-dimensional Monte Carlo integration.

2.1.5 Importance sampling

So far we have considered methods for generating samples directly from some target density. Although samples can be of value in themselves for giving a representative set of plausible values from the target density (e.g. for visualisation purposes), usually the end goal is to estimate integrals of the form in (2.1).

Importance sampling [53] is a Monte Carlo method which allows arbitrary integrals to be estimated. If q is density of a probability measure which is absolutely continuous to the probability measure defined by the target density p (which requires that $p(\boldsymbol{x}) > 0 \Rightarrow q(\boldsymbol{x}) > 0$), then importance sampling is based on the identity

$$\bar{f} = \int_X f(\boldsymbol{x})\,p(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) = \frac{\int_X f(\boldsymbol{x})\,\frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}\,q(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})}{\int_X \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}\,q(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})}. \tag{2.15}$$

Each of the numerator and denominator in (2.15) take the form of an expectation of a measurable function of a random variable $\mathbf{x}$ with probability density $\mathrm{p}_{\mathbf{x}} = q$. Further the denominator is exactly equal to $Z = \int_X \tilde{p}(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})$. We therefore have that

$$Z\bar{f} = \mathbb{E}[w(\mathbf{x})f(\mathbf{x})] \;\text{ and }\; Z = \mathbb{E}[w(\mathbf{x})] \;\text{ with }\; w(\boldsymbol{x}) = \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})}. \tag{2.16}$$

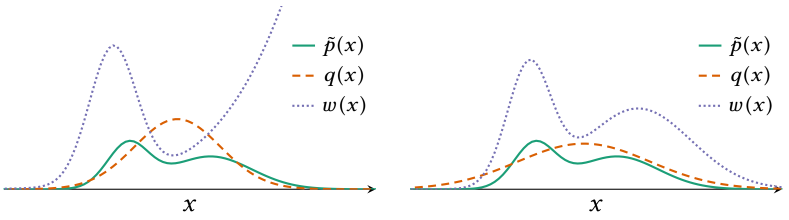


Figure 2.5: Visualisation of importance sampling. On both axes the green curve shows the unnormalised target density $\tilde{p}$, the dashed orange curve the density q values are sampled from and the dotted violet curve the importance weighting function $w(x) = \frac{\tilde{p}(x)}{q(x)}$ to estimate expectations with respect to the target density using samples from q. In the left axis the q chosen is undispersed compared to $\tilde{p}$ leading to very large w values in the right tail. In constrast in the right axis, the broader q leads to less extreme variation in w.

If we can generate random variables $\{\mathbf{x}_n\}_{n=1}^N$ each with marginal density q we can therefore form Monte Carlo estimates of both the numerator and denominator. We define $\hat{\mathrm{Z}}_N$ and $\hat{\mathrm{g}}_N$ as

$$\hat{\mathrm{Z}}_N = \frac{1}{N}\sum_{n=1}^{N} w(\mathbf{x}_n) \quad \text{and} \quad \hat{\mathrm{g}}_N = \frac{1}{\hat{\mathrm{Z}}}\sum_{n=1}^{N} w(\mathbf{x}_n) f(\mathbf{x}_n). \tag{2.17}$$

By the same argument as Section 2.1.1, $\mathbb{E}\left[\hat{\mathrm{Z}}_N\right] = Z$ and $\mathbb{E}[\hat{\mathrm{g}}_N] = Z\bar{f}$. We can therefore use importance sampling to form an unbiased estimate of the unknown normalising constant Z.

If we define $\hat{\mathrm{f}}_N = \hat{\mathrm{g}}_N / \hat{\mathrm{Z}}_N$, then this is a biased[4] estimator for $\bar{f}$ as in general the expectation of the ratio of two random variables is not equal to the ratios of their expectations. However if both the numerator and denominator have finite variance, i.e. $\mathbb{V}\left[\hat{\mathrm{Z}}_N\right] < \infty$ and $\mathbb{V}[\hat{\mathrm{g}}_N] < \infty$, then $\hat{\mathrm{f}}_N$ is a *consistent* estimator for $\bar{f}$ i.e. $\lim_{N\to\infty} \mathbb{E}\left[\hat{\mathrm{f}}_N\right] = \bar{f}$.

The $w(\mathrm{x}_n)$ values are typically termed the *importance weights*. If a few of the weights are very large, the weighted sums in (2.17) will be dominated by those few values, reducing the effective number of samples in the Monte Carlo estimates. This can particularly be a problem if the are regions of X with low probability under q where $p(\boldsymbol{x}) \gg q(\boldsymbol{x})$. As sampling points in these regions will be a rare event, a large number of samples may be needed to diagnose the issue adding further difficulty.

[4] In cases where the normalising constant Z is known, we can instead use $w(\boldsymbol{x}) = \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$ in which case the ratio estimator is not required and an unbiased estimates can be calculated. As the problems we are interested in will generally have unknown Z we do not consider this case further

A general recommendation is to use densities q with tails as least as heavy of thos of p, and in general the closer the match between q and p the better [66, 84]. Figure 2.5 shows a visualisation of the effect of the choice of q on the importance weights.

A heuristic that can be used to help assess the quality of importance sampling estimates is what is sometimes termed the *effective sample size* [60, 84]. It approximately quantifies how many independent samples from the target p would be required to get a Monte Carlo estimate with a similar variance to that achieved using an importance sampling estimator with weights $\{w(\mathbf{x}_n)\}_{n=1}^N$ given iid $\{\mathbf{x}_n\}_{n=1}^N$ generated from q. It can be calculated as

$$\mathsf{N}_{\text{eff}} = \left(\sum_{n=1}^{N} \bar{\mathsf{w}}_n^2\right)^{-1} \quad \text{where} \quad \bar{\mathsf{w}}_n = \frac{w(\mathbf{x}_n)}{\sum_{m=1}^N w(\mathbf{x}_m)}, \tag{2.18}$$

i.e. as the reciprocal of the sum of squares of the normalised importance weights. If $\mathsf{N}_{\text{eff}} \ll N$ this can suggest an issue with the choice of sampling density q. The diagnostic is not fool proof however as it is based on a finite sample size, and it may be that rare extreme importance weights due to e.g. a mode of the target p in the tails of q, are not encountered in a particular run giving a misleadingly high N_{eff}.

When previously discussing rejection sampling, we introduced an auxiliary binary accept indicator variable, a, associated with each proposed sample $\mathbf{x}$ (see Figure 2.4). If we generate N independent proposal – indicator pairs $\{\mathbf{x}_n, \mathsf{a}_n\}_{n=1}^N$ then the number of accepted proposals is $\mathsf{N}_{\text{acc}} = \sum_{n=1}^N \mathsf{a}_n$. Conditioned on N_{acc} being a value more than one, the generated rejection sampling variables $\{\mathbf{x}_n, \mathsf{a}_n\}_{n=1}^N$ can be used to form an *unbiased* Monte Carlo estimate of $\bar{f}$ using the estimator

$$\hat{\mathsf{f}}_N^{\text{RS}} = \frac{\sum_{n=1}^N \mathsf{a}_n f(\mathbf{x}_n)}{\sum_{m=1}^N \mathsf{a}_m}, \tag{2.19}$$

which just correponds to computing the empirical mean of the accepted proposals i.e. the standard Monte Carlo estimator. In comparison importance sampling forms a biased but consistent estimator for $\bar{f}$ from N samples $\{\mathbf{x}_n\}_{n=1}^N$ from a density q using the estimator

$$\hat{\mathsf{f}}_N^{\text{IS}} = \frac{\sum_{n=1}^N w(\mathbf{x}_n) f(\mathbf{x}_n)}{\sum_{m=1}^N w(\mathbf{x}_m)}. \tag{2.20}$$

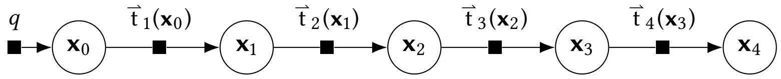

q
$\mathbf{x}_0$
$\vec{t}_1(\mathbf{x}_0)$
$\mathbf{x}_1$
$\vec{t}_2(\mathbf{x}_1)$
$\mathbf{x}_2$
$\vec{t}_3(\mathbf{x}_2)$
$\mathbf{x}_3$
$\vec{t}_4(\mathbf{x}_3)$
$\mathbf{x}_4$

Figure 2.6: Markov chain factor graph. The initial state $\mathbf{x}_0$ is sampled from a density q and each subsequent state $\mathbf{x}_n$ is then generated from a transition density $\vec{t}_n$ conditioned on the previous state $\mathbf{x}_{n-1}$.

From this perspective the accept indicators a_n in rejection sampling can be seen to act like binary importance weights, in contrast importance sampling using 'soft' weights which mean all sampled $\mathbf{x}_n$ make a contribution to the estimator (assuming $w(\boldsymbol{x}) \neq 0 \; \forall \boldsymbol{x} \in X$). However this correspondence is only loose. The rejection sampling estimator $\hat{f}_N^{\text{RS}}$ is unbiased unlike $\hat{f}_N^{\text{IS}}$, but this unbiasedness relies on conditioning on a non-zero value for N_{acc} (i.e. the number of accepted samples to generate) and continuing to propose points until this condition is met. In contrast importance sampling generates a fixed number of samples from q and does not use any auxiliary random variables.

Unlike rejection sampling, there is no need in importance sampling for q to upper-bound the target density (when multiplied by a constant). This allows more freedom in the choice of q however it is still important to choose q to be as close as possible to the target while remaining tractable to generate samples from. In general for target densities defined on high-dimensional spaces, it can be difficult to find an appropriate q such that the variation in importance weights is not too extreme [66]. As we will see later however the importance sampling framework can be combined with other methods we will discuss in the following sections to allow it to be scaled to high dimensional problems.

2.1.6 Markov chain Monte Carlo

The sampling methods we have considered so far have involved using independent random variables to form Monte Carlo estimates. However when introducing the Monte Carlo method we saw that is was not necessary for the random variables used in a Monte Carlo estimator to be independent. While it can be impractically computationally expensive to generate independent samples from complex high-dimensional target distributions, simulating a stochastic process which converges in distribution to the target and produces a sequence of *dependent* random variables is often a more tractable task. This is the idea exploited by *Markov chain Monte Carlo* (MCMC) methods.

A *Markov chain* is an ordered sequence of random variables $\{\mathbf{x}_n\}_{n=0}^{N}$ which have the *Markov property* — for all $n \in \{1 \dots N\}$, $\mathbf{x}_n$ is conditionally independent of $\{\mathbf{x}_n\}_{m<n-1}$ given $\mathbf{x}_{n-1}$. This conditional independence structure is visualised as a factor graph in Figure 2.6.

For a Markov chain defined on a general measurable state space $(X, \mathcal{F})$, the probability distribution of a state $\mathbf{x}_n$ given the state $\mathbf{x}_{n-1}$ is specified for each $n \in \{1 \dots N\}$ by a *transition operator*, $\vec{\mathsf{T}}_n : \mathcal{F} \times X \to [0, 1]$. In particular the transition operators define a series of regular conditional distributions for each $n \in \{1 \dots N\}$

$$\mathsf{P}_{\mathbf{x}_n}(A \,|\, \mathbf{x}_{n-1} = \boldsymbol{x}) = \vec{\mathsf{T}}_n(A \,|\, \boldsymbol{x}) \quad \forall A \in \mathcal{F},\ \boldsymbol{x} \in X. \tag{2.21}$$

We will generally assume that the chain is *homogeneous*, i.e. that the same transition operator is used for all steps $\vec{\mathsf{T}}_n = \vec{\mathsf{T}}\ \forall n \in \{1 \dots N\}$.

The key requirement of a transition operator for MCMC is that the target distribution P is *invariant* under the transition, that is it satisfies

$$P(A) = \int_X \vec{\mathsf{T}}(A \,|\, \boldsymbol{x})\, \mathrm{d}P(\boldsymbol{x}) \quad \forall A \in \mathcal{F}, \tag{2.22}$$

The invariance property means that if a chain state $\mathbf{x}_n$ is distributed according to the target P, all subsequent chain states $\mathbf{x}_{n+1}$, $\mathbf{x}_{n+2} \dots$ will also be marginally distributed according to the target. Therefore given a single random sample $\mathbf{x}_0$ from the target distribution, a series of dependent states marginally distributed according to the target could be generated and used to form Monte Carlo estimates of expectations.

Being able to generate even one exact sample from a complex high-dimensional target distribution is in general infeasible. Importantly however, subject to further conditions on the transition operator, the marginal distribution on the chain state $\mathsf{P}_{\mathbf{x}_n}$ of a Markov chain with a transition operator which leaves the target distribution invariant will converge to the target distribution irrespective of the distribution of the intial chain state. The requirements correspond to ensuring the target distribution is the *unique* invariant distribution of the chain.

To have a unique invariant distribution, a chain must be *irreducible* and *aperiodic* [119]. For a chain on a measurable space $(X, \mathcal{F})$, irreducibility is defined with respect to a measure ν, which could but does not necessarily need to be the target distribution P. A chain is ν-irreducible if starting at any point in X there is a non-zero probability of moving to any set with positive ν-measure in a finite number of steps, i.e.

$$\forall \boldsymbol{x} \in X,\, A \in \mathcal{F} : \nu(A) > 0 \;\; \exists\, m \in \mathbb{Z}^{+} : \mathsf{P}_{\mathbf{x}_m}(A \,|\, \mathbf{x}_0 = \boldsymbol{x}) > 0. \tag{2.23}$$

A chain with invariant distribution P is periodic if disjoint regions of the state space are visited cyclically, i.e. there exists an integer $r > 1$ and an ordered set of r disjoint P-positive subsets of X, $\{A_i\}_{i=1}^{r}$ such that $\vec{\mathsf{T}}(A_j \,|\, \boldsymbol{x}) = 1 \; \forall \boldsymbol{x} \in A_i,\; i \in \{1 \dots r\},\; j = (i+1) \mod r$.

If we can construct a ν-irreducible and aperiodic Markov chain $\{\mathbf{x}_n\}_{n=0}^{N}$ which has the target distribution P as its invariant distribution, then the MCMC *estimator* $\hat{\mathsf{f}}_N = \frac{1}{N}\sum_{n=1}^{N} f(\mathbf{x}_n)$ converges almost surely as $N \to \infty$ to $\bar{f} = \int_X f \,\mathrm{d}P$ for all starting states except for a ν-null set[5][74]. This convergence of *time-averages* (i.e. over states at different steps of the Markov chain) to *space-averages* (i.e. with respect to the stationary distribution across the state space), is termed *ergodicity* and is a consequence of the *Birkhoff–Khinchin ergodic theorem* [11].

Although irreducibility and aperiodicity of a Markov chain which leaves the target distribution invariant are sufficient for convergence of MCMC estimators, this does not tell us anything about the rate of that convergence and so how to quantify the error introduced by computing estimates with a Markov chain simulated for only a finite number of steps. Stronger notions of ergodicity can be used to help quantify convergence; we will concentrate on *geometric ergodicity* here. We first define a notion of distance between two measures μ and ν on a measurable space $(X, \mathcal{F})$, the *total variation distance*, as

$$\|\mu - \nu\|_{\mathrm{TV}} = \sup_{A \in \mathcal{F}} |\mu(A) - \nu(A)|. \tag{2.24}$$

For a ν-irreducible and aperiodic chain with invariant distribution P our earlier vague statement that the distribution on the chain state converges to P can now be restated more precisely as that for ν-almost all initial states $\mathbf{x}_0 = \boldsymbol{x}$, $\lim_{n\to\infty} \|\mathsf{P}_{\mathrm{x}^{(n)}}(\cdot \,|\, \mathbf{x}_0 = \boldsymbol{x}) - P\|_{\mathrm{TV}} = 0$. Geometric ergodicity makes a stronger statement that the convergence in total variation distance conditioned is geometric in n, i.e. that

$$\|\mathsf{P}_{\mathrm{x}^{(n)}}(\cdot \,|\, \mathbf{x}_0 = \boldsymbol{x}) - P\|_{\mathrm{TV}} \le M(\boldsymbol{x}) r^{n} \tag{2.25}$$

[5] The ‘except for a ν-null set’ caveat can be removed by requiring the stronger property of *Harris recurrence* [44].

for a positive measurable function M which depends on the initial chain state $\boldsymbol{x}$ and rate constant $r \in [0, 1)$. For chains which are geometrically ergodic, we can derive an expression for the *asymptoptic variance* of an MCMC estimator $\hat{\mathsf{f}}_N$ related to the variance of a simple Monte Carlo estimator previously considered in Section 2.1.1.

A stochastic process is stationary if the joint distribution of the states at any set of time points does not change if all those times are shifted by a constant.

As in Section 2.1.1 we define $\mathsf{f}_n = f(\mathbf{x}_n)$ and $\hat{\mathsf{f}}_N = \frac{1}{N}\sum_{n=1}^{N} \mathsf{f}_n$, with here the $\{\mathbf{x}_n\}_{n=1}^{N}$ the states of a Markov chain. For a homogeneous Markov chain with a unique invariant distribution P which is *stationary*, the marginal density on the states $\mathsf{P}_{\mathbf{x}_n}$ is equal to P for all n and we can use the expression for the variance of a general Monte Carlo estimator (which did not assume independence of the random variables) stated earlier in (2.4). Further the stationarity of the chain means that the covariance $\mathbb{C}[\mathsf{f}_n, \mathsf{f}_m]$ depends only on the difference $n - m$, and so the variance of the estimator simplifies to

$$\mathbb{V}\left[\hat{\mathsf{f}}_N\right] = \frac{\mathbb{V}[\mathsf{f}]}{N}\left(1 + 2\sum_{n=1}^{N-1}\left(\frac{N-n}{N}\frac{\mathbb{C}[\mathsf{f}_0, \mathsf{f}_n]}{\mathbb{V}[\mathsf{f}]}\right)\right). \tag{2.26}$$

If we multiply both sides of (2.26) by N and define $\rho_n = \frac{\mathbb{C}[\mathsf{f}_0, \mathsf{f}_n]}{\mathbb{V}[\mathsf{f}]}$ (the lag n autocorrelations of f), under the assumption that $\sum_{n=1}^{\infty} |\rho_n| < \infty$ in the limit of $N \to \infty$ we have that

$$\lim_{N\to\infty}\left(N\,\mathbb{V}\left[\hat{\mathsf{f}}_N\right]\right) = \mathbb{V}[\mathsf{f}]\left(1 + 2\sum_{n=1}^{\infty}\rho_n\right). \tag{2.27}$$

Now considering a chain which is geometrically ergodic from its initial state, if $\mathbb{E}\left[|\mathsf{f}|^{2+\delta}\right]$ is finite for some $\delta > 0$ then it can be shown [20, 36, 99] that (2.27) is also the asymptoptic variance for a *MCMC* estimator calculated using the chain states.

This motivates a definition of the *effective sample size* (ESS)[6] for an MCMC estimator $\hat{\mathsf{f}}_N$ computed using a geometrically ergodic chain as

$$N_{\text{eff}} = \frac{N}{1 + 2\sum_{n=1}^{\infty}\rho_n}. \tag{2.28}$$

The ESS quantifies the number of independent samples that would be required in a Monte Carlo estimator to give an equivalent variance to the MCMC estimator $\hat{\mathsf{f}}_N$ *in the asymptoptic limit* $N \to \infty$. In practice we cannot evaluate the exact autocorrelations and so we can only compute

[6] Note this is unrelated to the previous definition for importance sampling.

an estimated ESS, $\hat{N}_{\text{eff}}$, from one or more simulated chains with the estimation method needing to be carefully chosen to ensure reasonable values [118]. Although the assumption of geometric ergodicity can often be hard to verify in practice and ESS estimates can give misleading results in chains far from convergence, when used appropriately estimated ESSs can still be a useful heuristic for evaluating and comparing the effiency of Markov chain estimators and are often available as a standard diagnostic in MCMC software packages [18, 91, 106].

So far we have not discussed how to actually construct a transition operator giving a chain with the required invariant distribution. As a notational convenience we will consider the transition operator as being specified by a *transition density* $\vec{\mathrm{t}} : X \times X \to [0, \infty)$ which is defined with respect to a base measure μ (which we will later assume to be the same as that which the target density we wish to integrate against is defined with respect to, hence the reuse of notation). The transition operator is then

$$\vec{\mathrm{T}}(A \,|\, \boldsymbol{x}) = \int_A \vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}') \quad \forall A \in \mathcal{F},\ \boldsymbol{x} \in X. \tag{2.29}$$

In practice the probability measure defined by a transition operator will often have a singular component, for example corresponding to a non-zero probability of the chain remaining in the current state. In this case $\vec{\mathrm{T}}$ is not absolutely continuous with respect to μ and the transition density is not strictly defined. As we did in the previous chapter however we will informally use Dirac deltas to represent a 'density' of singular measures, and so still consider a transition density as existing. The requirement that the transtion operator leaves the target distribution invariant, can then be expressed in terms of the target density p and transition density $\vec{\mathrm{t}}$ as

$$p(\boldsymbol{x}') = \int_X \vec{\mathrm{t}}\,(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}) \quad \forall \boldsymbol{x}' \in X. \tag{2.30}$$

Finding a transition density which leaves the target density invariant by satisfying (2.30) seems difficult in general as it involves evaluating an integral against the target density - precisely the computational task which we have been forced to seek approximate solutions to. We can make progress by considering the joint density of a pair of successive states for a chain with invariant distribution P that has converged to stationarity. Then we have that

$$\mathsf{p}_{\mathbf{x}_n,\mathbf{x}_{n-1}}(\boldsymbol{x}',\boldsymbol{x}) = \mathsf{p}_{\mathbf{x}_n\,|\,\mathbf{x}_{n-1}}(\boldsymbol{x}'\,|\,\boldsymbol{x})\,\mathsf{p}_{\mathbf{x}_{n-1}}(\boldsymbol{x}) = \overrightarrow{\mathsf{t}}(\boldsymbol{x}'\,|\,\boldsymbol{x})\,p(\boldsymbol{x}). \tag{2.31}$$

We can also consider factorising this joint density into the product of the marginal density of the current state $\mathsf{p}_{\mathbf{x}_n}$ and the conditional density of the previous state given the current state $\mathsf{p}_{\mathbf{x}_{n-1}|\mathbf{x}_n}$. Due to stationarity $\mathsf{p}_{\mathbf{x}_n}$ is also equal to p and so we have that $\mathsf{p}_{\mathbf{x}_{n-1}|\mathbf{x}_n}$ must be the density of a transition operator which also leaves P invariant, corresponding to a time reversed version of the original (stationary) Markov chain[7]. If we therefore denote $\overleftarrow{\mathsf{t}} = \mathsf{p}_{\mathbf{x}_{n-1}|\mathbf{x}_n}$ (and which we will term the *backward transition density* in contrast to $\overrightarrow{\mathsf{t}}$ which in this context we will qualify as the *forward transition density*), we have that

$$\overrightarrow{\mathsf{t}}(\boldsymbol{x}'\,|\,\boldsymbol{x})\,p(\boldsymbol{x}) = \overleftarrow{\mathsf{t}}(\boldsymbol{x}\,|\,\boldsymbol{x}')\,p(\boldsymbol{x}') \quad \forall \boldsymbol{x} \in X,\; \boldsymbol{x}' \in X. \tag{2.32}$$

Integrating both sides with respect to $\boldsymbol{x}$, we have that $\forall \boldsymbol{x}' \in X$

$$\begin{aligned}\int_X \overrightarrow{\mathsf{t}}(\boldsymbol{x}'\,|\,\boldsymbol{x})\,p(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) &= \int_X \overleftarrow{\mathsf{t}}(\boldsymbol{x}\,|\,\boldsymbol{x}')\,p(\boldsymbol{x}')\,\mathrm{d}\mu(\boldsymbol{x}) \\ &= \int_X \overleftarrow{\mathsf{t}}(\boldsymbol{x}\,|\,\boldsymbol{x}')\,\mathrm{d}\mu(\boldsymbol{x})\,p(\boldsymbol{x}') = p(\boldsymbol{x}'),\end{aligned} \tag{2.33}$$

and so that (2.30) is satisfied, with the last inequality arising due to $\overleftarrow{\mathsf{t}}$ being a normalised density on its first argument. Therefore if we can find a pair of transition densities, $\overrightarrow{\mathsf{t}}$ and $\overleftarrow{\mathsf{t}}$, satisfying (2.32), then the transition operator specified by $\overrightarrow{\mathsf{t}}$ will leave the target distribution P invariant (and by an equivalent argument so will the transition operator specified by $\overleftarrow{\mathsf{t}}$). We can further simplify (2.32) by requiring that $\overrightarrow{\mathsf{t}} = \overleftarrow{\mathsf{t}} = \mathsf{t}$, i.e. that both forward and backward transition densities (and corresponding operators) take the same form and so that the chain at stationarity is *reversible*, in which case have that

$$\mathsf{t}(\boldsymbol{x}'\,|\,\boldsymbol{x})\,p(\boldsymbol{x}) = \mathsf{t}(\boldsymbol{x}\,|\,\boldsymbol{x}')\,p(\boldsymbol{x}') \quad \forall \boldsymbol{x} \in X,\; \boldsymbol{x}' \in X. \tag{2.34}$$

This is often termed the *detailed balance* condition. Importantly both the detailed balance (2.34) and *generalised balance* (2.32) conditions can

[7] The time reversal of a Markov chain is always itself a a Markov chain irrespective of stationarity (as the defining conditional independence structure is symmetric with respect to the direction of time), however the reverse of a homogeneous Markov chain which is not stationary will not in general itself be homogeneous.

also be written in terms of the unormalised density $\tilde{p}$ by multiplying both sides by Z, and so can be checked even when Z is unknown.

The restriction to reversible transition operators in detailed balance, while sufficient for (2.30) to hold is not necessary. Markov chains which satisfy the generalised balance condition but not detailed balance are termed *non-reversible*, and there are theoretical results suggesting that non-reversible Markov chains can sometimes achieve significantly improved convergence compared to related reversible chains [24, 50, 80] (a criticism made of some of these results is that choice of 'reference' reversible chain to compare to can be somewhat arbitrary[78]).

While there are several general purpose frameworks for specifying reversible transition operators which leave a target distribution invariant, developing methods for constructing irreversible transition operators with a desired invariant distribution has proven more challenging. The approaches proposed to date are generally limited in practice to special cases such as finite state spaces [114, 115, 120] or chains with tractable invariant distributions such as multivariate normal [10].

Nonetheless non-reversible Markov chains are still commonly used in MCMC applications. Given a set of transition operators which each individually leave a target distribution invariant, the sequential composition of the transition operators will necessarily by induction also leave the target distribution invariant. Even if the individual transition operators are all reversible, the overall sequential composition will generally not be (instead having an adjoint 'backward' operator corresponding to applying the individual transitions in the reversed order). Sequentially combining several reversible transition operators is common in MCMC implementations, though this is more often the result of each individual operator not meaning the requirements for ergodicity in isolation and so needing to be combined with other operators, rather than due to a specific aim of introducing irreversibility.

Having now introduced the key theoretical concepts underlying MCMC methods, we will now move on to discussing details of their implementation. In the following sub-sections we will review three widely applicable frameworks for constructing reversible transition operators which leave a target distribution invariant: the *Metropolis–Hastings* algorithm, *Gibbs sampling* and *slice sampling*. These three methods will form the basis for much of the work introduced in this thesis.

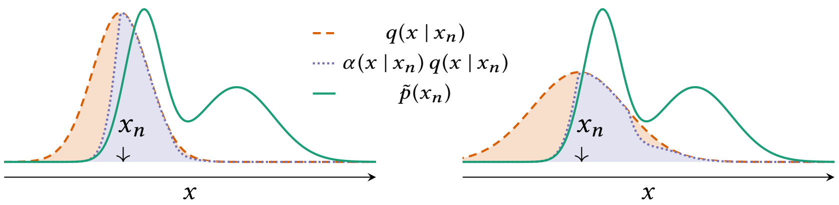


Figure 2.7: Visualisation of Metropolis–Hastings algorithm in a univariate target density. The green curves shows the unnormalised target density. The arrows indicate the current chain state. The orange curves show the density of proposed moves from this state, with the left axis using a narrower proposal than the right. The violet curves show the proposal density scaled by the acceptance probability of the proposed move, this reducing the probability of transitions to states with lower density than the current state. The orange region between the violet and orange curves represents the probability mass reallocated to rejections by the downscaling by the acceptance function. The broader proposal in the right axis has an increased probability of making a move to the other mode in the target density but at a cost of an increased rejection probability.

Algorithm 3 Metropolis–Hastings transition.

Input: $\boldsymbol{x}_n$: current chain state, $\tilde{p}$: unnormalised target density,
q : normalised proposal density which we can sample from.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
$\boldsymbol{x}^* \sim q(\boldsymbol{x}_n)$
$u \sim \mathcal{U}(0,1)$
if $u < \frac{\tilde{p}(\boldsymbol{x}^*)\, q(\boldsymbol{x} \mid \boldsymbol{x}^*)}{\tilde{p}(\boldsymbol{x})\, q(\boldsymbol{x}^* \mid \boldsymbol{x})}$ then
    $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}^*$
else
    $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_n$
return $\boldsymbol{x}_{n+1}$
```

2.1.6.1 *Metropolis–Hastings*

The seminal *Metropolis–Hastings* algorithm provides a general framework for constructing Markov chains with a desired invariant distribution and is ubitiqous in MCMC methodology. The original Rosenbluth–Teller–Metropolis variant of the algorithm [73] dates to the very beginnings of the Monte Carlo method, having being first implemented on Los Alamos' MANIAC[8] one of the earliest programmable computers. The method was generalised in a key paper by Hastings [46], and the optimality among several competing alternatives of the form now used demonstrated by Peskun [88]. An extension to Markov chains on trans-dimensional spaces was proposed by Green [41].

Although the algorithm has come to be commonly known by Edward Metropolis' name as first author on the 1953 paper [73], it is believed that Arianna and Marshall Rosenbluth, two of the other co-authors, were the main contributors to the development of the algorithm [42].

[8] *Mathematical Analyzer, Numerical Integrator and Computer.*

An outline of the method is given in Algorithm 3 and a visualisation of terms involved in the transition operator in a univariate target density shown in Figure 2.7. The key idea is to propose updates to the state using an arbitrary transition operator, and then correct for this transition operator not necessarily leaving the target distribution invariant by stochastically accepting or rejecting the proposal. If a proposal is rejected the chain remains at the current state, otherwise the chain state takes on the proposed value.

The transition density corresponding to Algorithm 3 is

$$\begin{aligned}\mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \alpha(\boldsymbol{x}' \,|\, \boldsymbol{x})\, q(\boldsymbol{x}' \,|\, \boldsymbol{x}) + \\ \left(1 - \int_X \alpha(\boldsymbol{x}^* \,|\, \boldsymbol{x})\, q(\boldsymbol{x}^* \,|\, \boldsymbol{x})\, \mathrm{d}\mu(\boldsymbol{x}^*)\right) \delta(\boldsymbol{x}' - \boldsymbol{x}),\end{aligned} \tag{2.35}$$

with the *acceptance probability* $\alpha : X \times X \to [0, 1]$ defined as

$$\alpha(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \min\left\{1, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}\right\} = \min\left\{1, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, \tilde{p}(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, \tilde{p}(\boldsymbol{x})}\right\}, \tag{2.36}$$

and $q : X \times X \to [0, \infty)$ the density of the proposal transition operator (from herein *proposal density*). This transition density corresponds to a reversible transition operator which leaves the target distribution P invariant as we will now show.

For the purposes of verifying the detailed balance condition (2.34), the density of *self-transitions*, i.e. a transition to the same state, can be ignored as (2.34) is trivially satisfied for $\boldsymbol{x}' = \boldsymbol{x}$. Considering therefore the cases $\boldsymbol{x} \neq \boldsymbol{x}'$ where the Dirac delta term representing the singular measure corresponding to rejected proposals can be neglected, we therefore have $\forall \boldsymbol{x} \in X,\, \boldsymbol{x}' \in X : \boldsymbol{x} \neq \boldsymbol{x}$

$$\begin{aligned}\mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}) &= \min\left\{1, \frac{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')}{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}\right\} q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}) &(2.37)\\ &= \min\{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x}),\, q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')\} &(2.38)\\ &= \min\left\{\frac{q(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})}{q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}')}, 1\right\} q(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}') &(2.39)\\ &= \mathrm{t}(\boldsymbol{x} \,|\, \boldsymbol{x}')\, p(\boldsymbol{x}'). &(2.40)\end{aligned}$$

Therefore the detailed balance condition is satisfied, and the Metropolis–Hastings transition operator leaves the target distribution P invariant.

The original Rosenbluth–Teller–Metropolis algorithm used a symmetric proposal density $q(\boldsymbol{x}' \,|\, \boldsymbol{x}) = q(\boldsymbol{x} \,|\, \boldsymbol{x}')\ \forall \boldsymbol{x} \in X, \boldsymbol{x}' \in X$ (with the extension to the non-symmetric case being due to Hastings), in which case the acceptance probability definition simplifies to

$$\alpha(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \min\left\{1, \frac{p(\boldsymbol{x}')}{p(\boldsymbol{x})}\right\} = \min\left\{1, \frac{\tilde{p}(\boldsymbol{x}')}{\tilde{p}(\boldsymbol{x})}\right\}. \tag{2.41}$$

Note that importantly in both (2.36) and (2.41) the target density only appears as a ratio and so the density need only be known up to a proportionality constant.

An important special case for chains on a Euclidean state space with a Borel σ-algebra i.e. $(X, \mathcal{F}) = (\mathbb{R}^D, \mathscr{B}(\mathbb{R}^D))$, is when the proposal transition operator is deterministic and corresponds to a differentiable involution of the current state. Let $\boldsymbol{\phi} : X \to X$ be an involution, i.e. $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}\ \forall X$ with Jacobian determinant $J_{\boldsymbol{\phi}}(\boldsymbol{x}) = \left|\frac{\partial \boldsymbol{\phi}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|$ which is defined and non-zero P-almost everywhere. Then if we define a transition operator via the transition density

$$\begin{aligned} \mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x}) &= \delta(\boldsymbol{x}' - \boldsymbol{\phi}(\boldsymbol{x}))\alpha(\boldsymbol{x}) + \delta(\boldsymbol{x}' - \boldsymbol{x})(1 - \alpha(\boldsymbol{x})), \\ \alpha(\boldsymbol{x}) &= \min\left\{1, \frac{p \circ \boldsymbol{\phi}(\boldsymbol{x})}{p(\boldsymbol{x})} J_{\boldsymbol{\phi}}(\boldsymbol{x})\right\}, \end{aligned} \tag{2.42}$$

then this transition operator will leave the target distribution P invariant. This deterministic transition operator variant can be viewed as a special case of the trans-dimensional Metropolis–Hastings extension introduced by Green [37, 41]. To generate from this transition operator from a current state $\boldsymbol{x}$ we compute the proposed move $\boldsymbol{\phi}(\boldsymbol{x})$ and accept the move with probablity $\alpha(\boldsymbol{x})$. We can demonstrate that this transition operator leaves P invariant by directly verifying (2.30)

$$\int_X \mathrm{t}(\boldsymbol{x}' \,|\, \boldsymbol{x})\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \tag{2.43}$$

$$= \int_X \delta(\boldsymbol{x}' - \boldsymbol{\phi}(\boldsymbol{x}))\, \alpha(\boldsymbol{x})\, p(\boldsymbol{x}) + \delta(\boldsymbol{x}' - \boldsymbol{x})(1 - \alpha(\boldsymbol{x}))\, p(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \tag{2.44}$$

$$= \int_X \delta(\boldsymbol{x}' - \boldsymbol{y})\, \alpha \circ \boldsymbol{\phi}(\boldsymbol{y})\, p \circ \boldsymbol{\phi}(\boldsymbol{y})\, J_{\boldsymbol{\phi}}(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} + (1 - \alpha(\boldsymbol{x}'))\, p(\boldsymbol{x}') \tag{2.45}$$

$$= p(\boldsymbol{x}') + \alpha \circ \boldsymbol{\phi}(\boldsymbol{x}')\, p \circ \boldsymbol{\phi}(\boldsymbol{x}')\, J_{\boldsymbol{\phi}}(\boldsymbol{x}') - \alpha(\boldsymbol{x}')\, p(\boldsymbol{x}'). \tag{2.46}$$

In going from (2.44) to (2.45) we use a change of variables $\boldsymbol{y} = \boldsymbol{\phi}(\boldsymbol{x})$ in the integral. As $\boldsymbol{\phi}$ is an involution we have that $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}') = \boldsymbol{x}'$ and $J_{\boldsymbol{\phi}} \circ \boldsymbol{\phi}(\boldsymbol{x}') = J_{\boldsymbol{\phi}}(\boldsymbol{x}')^{-1}$ and so

$$\begin{aligned} \alpha \circ \boldsymbol{\phi}(\boldsymbol{x}')\, p \circ \boldsymbol{\phi}(\boldsymbol{x}')\, J_{\boldsymbol{\phi}}(\boldsymbol{x}') &= \min\Big\{ p \circ \boldsymbol{\phi}(\boldsymbol{x}')\, J_{\boldsymbol{\phi}}(\boldsymbol{x}'),\, p(\boldsymbol{x}') \Big\} \\ &= \alpha(\boldsymbol{x}')\, p(\boldsymbol{x}'). \end{aligned} \tag{2.47}$$

The last two terms in (2.46) therefore cancel and we have that (2.30) is satisfied by the transition operator defined by (2.42). Although this transition operator leaves the target distribution P invariant, it is clear that it will not generate an ergodic Markov chain. Starting from a point $\boldsymbol{x}$ the next chain state will be either $\boldsymbol{\phi}(\boldsymbol{x})$ if the proposed move is accepted or $\boldsymbol{x}$ if rejected. In the former case the next proposed move will be to $\boldsymbol{\phi} \circ \boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$ i.e. back to the original state. Therefore the chain will visit a maximum of two states. However as noted previously we can sequentially compose individual transition operators which all leave a target distribution invariant. Therefore a deterministic proposal Metropolis–Hastings transition can be combined with other transition operators to ensure the chain is irreducible and aperiodic.

In general for a Metropolis–Hastings transition operator to be irreducible, it is necessary that the proposal operator is irreducible [119], however this is not sufficient. For a target density which is positive everywhere on $X = \mathbb{R}^D$, then a sufficient but not necessary condition for irreducibility is that the proposal density is positive everywhere [99]. If the set of points with a non-zero probability of rejection has non-zero P-measure, then the transition operator is aperiodic [119].

A common choice of proposal density when the target distribution is defined on a $(\mathbb{R}^D, \mathscr{B}(\mathbb{R}^D))$ is a multivariate normal density centred at the current state i.e. $q(\boldsymbol{x}' \,|\, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}' \,|\, \boldsymbol{x}, \Sigma)$ which satisfies the positivity condition for irreducibility. In general we would expect improved performance with a proposal density covariance Σ which is proportional to the true covariance of the target distribution [103], in practice we do not have access to the true covariance and so typically an isotropic proposal density is used with covariance $\Sigma = \sigma^2\mathbf{I}$ controlled by a single scale parameter σ, often termed the *step size* or *proposal width*. This proposal density is symmetric so the simplified acceptance rule (2.41) can be used, further the prosal density depends only on the differ-

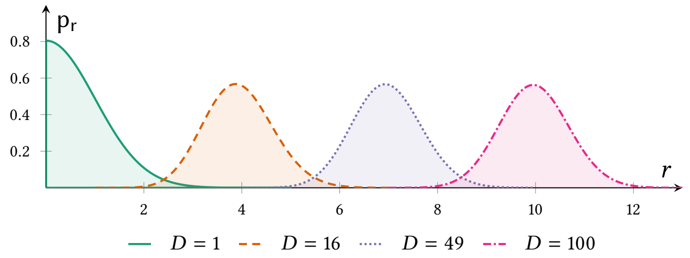


Figure 2.8: Illustration of concentration of measure in a multivariate normal distribution. The plots shows the probability density of the distance from the origin $\mathrm{r} = \|\mathbf{x}\|_2$ of a D-dimensional multivariate normal random vector $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ for different dimensionalities D. As the dimension increases most of the mass concentrates away from the origin around a spherical shell of radius $\sqrt{D}$. For a multivariate normal random vector with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ this generalises to the mass being mainly in an ellipsoidal shell aligned with the eigenvectors of $\boldsymbol{\Sigma}$ and centred at $\boldsymbol{\mu}$.

ence $\boldsymbol{x}' - \boldsymbol{x}$ with Metropolis–Hastings methods having these properties often termed *random-walk Metropolis*.

Random walk Metropolis methods have been extensively theoretically studied, with sufficient conditions known in some cases to ensure geometric ergodicity of a chain [72, 101] though these can be hard to verify in practical problems. There has also been much work on practical guidelines and methods for tuning the free parameters in the algorithm, including approaches for tuning the step-size using acceptance rates [33, 98] and adaptive variants which automatically estimate a non- isotropic proposal covariance [43, 103].

In general the choice of proposal density will be key in determining the efficiency of Metropolis–Hastings MCMC methods. Ideally we want to be able to propose large moves in the state space to reduce the depencies between successive chain states and so increase the number of effective samples, however this needs to be balanced with maintaining a reasonable acceptance probability with large proposed moves often having a low acceptance probability. Figure 2.7 gives an illustration of this tradeoff in a one-dimensional example.

In high-dimensional spaces this issue is much more severe due to the phenomenon of *concentration of measure*: in probability distributions defined on high-dimensional spaces most of the probability mass will tend to be concentrated into a ‘small’ subset of the space [4, 66]. An illustration of this phenomenon for the multivariate normal distribution

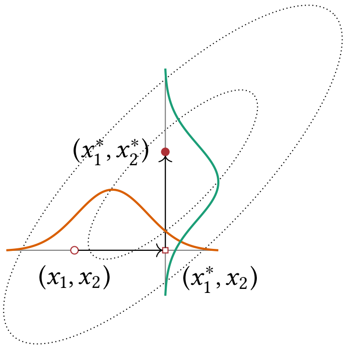


Figure 2.9: Schematic of Gibbs sampling transition in a bivariate normal target distribution (ellipses indicate constant density contours). Given an initial state $\mathbf{x} = (x_1, x_2)$, the x_1 (horizontal) co-ordinate is first updated by independently sampling from the normal conditional $\mathrm{p}_{\mathrm{x}_1 | \mathrm{x}_2}(\cdot \,|\, x_2)$, represented by the orange curve. The new partially updated state is then $\mathbf{x} = (x_1^*, x_2)$. The second x_2 (vertical) co-ordinate is then independently resampled from the normal conditional $\mathrm{p}_{\mathrm{x}_2 | \mathrm{x}_1}(\cdot \,|\, x_1^*)$, shown by the green curve. The final updated state is then $\mathbf{x} = (x_1^*, x_2^*)$.

is shown in Figure 2.8, where the mass in high dimensions is mostly located in a thin ellipsoidal shell. The region where most of the mass concentrates, termed the *typical set* of the distribution, will for the target distributions of interest generally have a significantly more complex geometry. Finding proposals which can make large moves in such settings is challenging: moves in most directions will have a probability of acceptance which exponentially drops to zero as the distance away from the current state is increased and so simple proposal densities which ignore the geometry the typical set such as those used in random-walk Metropolis will need to make very small moves to have a reasonable probability of acceptance [9].

To tackle this issue methods have been proposed which exploit more information about the target distribution than just the point evaluations of the density in the Metropolis–Hasting acceptance probability term. *Hamiltonian Monte Carlo* (HMC) methods, which make use of the *gradient* of the target density, are a particularly important class of such methods and a central focus of this thesis. We will discuss HMC methods in detail in Chapter 3.

2.1.6.2 Gibbs sampling

Gibbs sampling [32, 35], originally proposed by Geman and Geman for image restoration using a Markov random field image model, is based on the observation that a valid transition operator for a joint target

Algorithm 4 Sequential Gibbs transition.

Input: $\boldsymbol{x}_n$: current chain state, I : ordered index set over individual variables in chain state, $\{p_i\}_{i \in I}$: set of complete conditionals of target density p which can all be sampled from.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
$\boldsymbol{x} \leftarrow \boldsymbol{x}_n$
for $i \in I$ do
    $x_i \sim p_i(\boldsymbol{x}_{\setminus i})$
$\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}$
return $\boldsymbol{x}_{n+1}$
```

distribution across many variables, is one which updates only a subset of the variables and leaves the conditional distribution on that subset given the rest invariant. Although if used in isolation a transition operator which only updates some components of the state will not give an ergodic chain, as discussed previously multiple transition operators can be combined together to achieve ergodicity.

More specifically the original formulation of Gibbs sampling defines a Markov chain by sequentially independently resampling each individual variable in the model from its conditional distribution given the current values of the remaining variables. If I is an index set over the individual variables in the vector target state $\mathbf{x}$, then for each $i \in I$ we partition the state $\mathbf{x}$ into the i^{th} variable x_i and a vector containing all the remaining variable values $\mathbf{x}_{\setminus i}$. For each $i \in I$ the target density can be factorised in to the marginal density $p_{\setminus i}$ on $\mathbf{x}_{\setminus i}$ and conditional density p_i on x_i given $\mathbf{x}_{\setminus i}$, i.e.

$$p(\boldsymbol{x}) = p_i(x_i \,|\, \boldsymbol{x}_{\setminus i})\, p_{\setminus i}(\boldsymbol{x}_{\setminus i}), \tag{2.48}$$

with the conditional densities $\{p_i\}_{i \in I}$ termed the *complete conditionals* of the target density. If each of these complete conditionals corresponds to a distribution we can generate samples from (for example using a transform method or rejection sampling) then we can apply the sequential Gibbs sampling transition operator defined in Algorithm 4 and visualised for a bivariate example in Figure 2.9.

The sequential Gibbs transition is irreducible and aperiodic under mild conditions [19, 100]. Rather than using a deterministic sequential scan through the variables, a variant is to randomly sample without replacement the variable to update on each iteration; unlike the sequential scan version this defines a reversible transition operator. This random update variant is somewhat more amenable to theoretical analysis and comes with some stronger guarantees, however in practice the ease of implementation of the sequential scan variant and computational benefits in terms of memory access locality mean it is more often used in practice [47]. A compromise between the completely random updates and a sequential scan is to randomly permute the update order have each complete scan.

A apparent advantage of Gibbs sampling over Metropolis–Hastings is the lack of a proposal density which needs to be tuned. This is has helped popularise ‘black-box’ implementations of Gibbs sampling such as the probabilistic modelling packages BUGS [38] and JAGS [90]. A well-known issue with Gibbs sampling however is that its performance is highly dependent on the parameterisation used for the target density [92], with strong correlations between variables leading to large dependencies between successive states and slow convergence to stationarity. This can be alleviated in some cases by using a suitable reparameterisation to reduce dependencies between variables, however this restores the difficulty of tuning free parameters.

The updates do not necessarily need to be performed by sampling from complete conditions of single variables - in some cases the complete conditional of a vector variables has a tractable form which can be sampled from as a ‘block’; this motivates the name *block Gibbs samling* for such variants. By accounting for the dependencies between the variables in a block this can help alleviate some of the issues with highly correlated targets where applicable.

Compound constructions such as *Metropolis(–Hastings)-within-Gibbs* are sometimes used to refer to methods which sequentially apply *Metropolis–Hastings* transition operators which each update only a subset of variables in the target distribution. We will prefer to however consider the defining feature of Gibbs sampling as being exact sampling from one or more conditionals rather than sequentially applying transition operators which update only subsets of variables and so will only refer to ‘Gibbs sampling’ in that context.

2.1.6.3 Slice sampling

The final general class of MCMC methods we will consider is *slice sampling*.

Algorithm 5 Linear slice sampling transition.

Input: $\boldsymbol{x}_n$: current chain state, $\tilde{p}$: unnormalised target density,
q : slice vector density.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
$h \sim \mathcal{U}(0, \tilde{p}(\boldsymbol{x}_n))$
$b_u \sim \mathcal{U}(0, 1)$
$b_l \leftarrow b_u - 1$
$b \sim \mathcal{U}(b_l, b_u)$
$\boldsymbol{v} \sim q$
while TRUE do
    $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_n + b\boldsymbol{v}$
    if $\tilde{p}(\boldsymbol{x}^*) \leq h$ then
        if $b < 0$ then $b_l \leftarrow b$ else $b_u \leftarrow b$
        $b \sim \mathcal{U}(b_l, b_u)$
    else
        return $\boldsymbol{x}^*$
```

Algorithm 6 Elliptical slice sampling transition.

Input: $\boldsymbol{x}_n$: current chain state, $\tilde{p}$: unnormalised target density,
Σ : covariance of normal approximation to target.
Output: $\boldsymbol{x}_{n+1}$: next chain state with $\boldsymbol{x}_n \sim p \implies \boldsymbol{x}_{n+1} \sim p$.

```
$h \sim \mathcal{U}(0, \tilde{p}(\boldsymbol{x}_n) / \mathcal{N}(\boldsymbol{x}_n \,|\, \boldsymbol{0}, \Sigma))$
$\theta_u \sim \mathcal{U}(0, 2\pi)$
$\theta_l \leftarrow \theta_u - 2\pi$
$\theta \leftarrow \theta_u$
$\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \Sigma)$
while TRUE do
    $\boldsymbol{x}^* \leftarrow \boldsymbol{x}_n \cos\theta + \boldsymbol{v} \sin\theta$
    if $\tilde{p}(\boldsymbol{x}^*) / \mathcal{N}(\boldsymbol{x}^* \,|\, \boldsymbol{0}, \Sigma) \leq h$ then
        if $\theta < 0$ then $\theta_l \leftarrow \theta$ else $\theta_u \leftarrow \theta$
        $\theta \sim \mathcal{U}(\theta_l, \theta_u)$
    else
        return $\boldsymbol{x}^*$
```

2.2 OPTIMISATION APPROACHES

The central idea of the methods we will review in this section is to try to find a normalised probability density $q(\boldsymbol{x})$ from a 'simple' family that in some sense approximates the target density, i.e. $p(\boldsymbol{x}) \approx q(\boldsymbol{x})$. Depending on the family chosen for q, integrals of some functions f against the target density p, can be approximated by analytic solutions to integrals of f against q e.g. if $q(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \,|\, \boldsymbol{\mu}, \Sigma)$ then we can approximate the mean of the target density as $\boldsymbol{\mu}$ and the covariance as Σ. To compute integrals of more general functions f we will generally need to resort to using one of the sampling approaches we will review in the next section; generally it will be possible to directly generate independent samples from q while often this will not be the case for p hence this two-step approach still offers (computational) advantages over directly applying a sampling approach. Often the approaches we will discuss also allow estimation of the normalising constant Z which may be needed for model comparison.

2.2.1 Laplace's method

For target densities p defined with respect to a D-dimensional Lebesgue measure λ^D, a simple approach for computing a multivariate normal approximate density q to p is *Laplace's method.* Although not always strictly required, in general the method will work better for target densities with unbounded support, and more generally for targets which are as 'close to normal' as possible. Therefore a useful initial step will often be to apply a change of variables to the target density, such that the density on the transformed space has unbounded support, for example working with the density on the logarithm of a random variable with support only on positive values.

The key idea in Laplace's method is to form a truncated Taylor series approximation to the logarithm of the unnormalised target density

$$\begin{aligned}\log \tilde{p}(\boldsymbol{x}) \approx \log \tilde{p}(\boldsymbol{x}^*) &+ \boldsymbol{g}(\boldsymbol{x}^*)^\mathsf{T}(\boldsymbol{x} - \boldsymbol{x}^*) \\ &+ \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{H}(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*),\end{aligned} \tag{2.49}$$

where the *gradient* and *Hessian* of $\log \tilde{p}$ are defined respectively as

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{\partial \log \tilde{p}(\boldsymbol{x})}{\partial \boldsymbol{x}}^\mathsf{T} \quad \text{and} \quad \boldsymbol{H}(\boldsymbol{x}) = \frac{\partial^2 \log \tilde{p}(\boldsymbol{x})}{\partial \boldsymbol{x} \partial \boldsymbol{x}^\mathsf{T}}. \tag{2.50}$$

If the point $\boldsymbol{x}^*$ the expansion is formed around is chosen to be a (loca) maxima of $\log \tilde{p}$, which necessarily means that the gradient is zero, $\boldsymbol{g}(\boldsymbol{x}^*) = \boldsymbol{0}$, and the Hessian negative definite, $\boldsymbol{H}(\boldsymbol{x}^*) \prec 0$, then

A matrix $\boldsymbol{M} \in \mathbb{R}^{D\times D}$ is positive semi definite*, denoted $\boldsymbol{M} \succeq 0$, iff $\boldsymbol{x}^\mathsf{T}\boldsymbol{M}\boldsymbol{x} \geq 0$ $\forall \boldsymbol{x} \in \mathbb{R}^D$ and* positive definite*, denoted $\boldsymbol{M} \succ 0$, if the inequality is made strict. Corresponding definitions for a* negative semi definite *matrices, $\boldsymbol{M} \preceq 0$, and negative definite matrices, $\boldsymbol{M} \prec 0$, are formed by reversing the sign of the inequality.*

$$\log \tilde{p}(\boldsymbol{x}) \approx \log \tilde{p}(\boldsymbol{x}^*) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{H}(\boldsymbol{x}^*)(\boldsymbol{x} - \boldsymbol{x}^*). \tag{2.51}$$

Taking the exponential of both sides we therefore have that

$$\tilde{p}(\boldsymbol{x}) \approx \tilde{p}(\boldsymbol{x}^*) \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}(-\boldsymbol{H}(\boldsymbol{x}^*))(\boldsymbol{x} - \boldsymbol{x}^*)\Big). \tag{2.52}$$

This has the form of an unnormalised multivariate normal density with mean $\boldsymbol{x}^*$ and inverse covariance (precision) $-\boldsymbol{H}(\boldsymbol{x}^*)$.

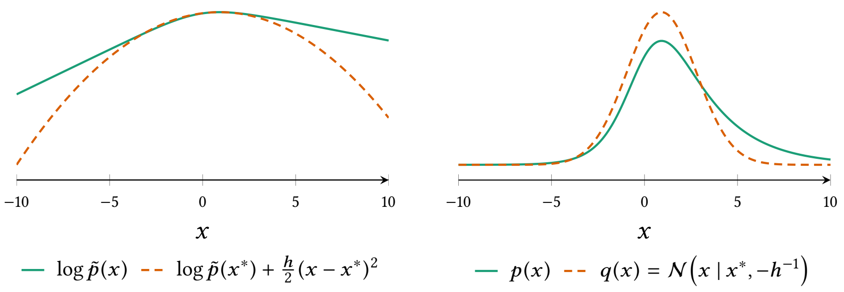


Figure 2.10: Univariate example of Laplace's method. Left axis shows the logarithm of the unnormalised target density $\log\tilde{p}(x)$ (green curve) and the corresponding quadratic Taylor series approximation $\log\tilde{p}(x^*) + \frac{h}{2}(x-x^*)^2$ (dashed orange curve) around the maxima x^* with $h = (\partial^2 \log\tilde{p}/\partial x^2)|_{x^*}$. The right axis shows the corresponding normalised target density $p(x)$ (green curve) and approximate density $q(x) = \mathcal{N}\left(x \mid x^*, -h^{-1}\right)$ (dashed orange curve).

This suggests setting the approximate density q to a multivariate normal density $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{x}^*, \boldsymbol{C})$ with $\boldsymbol{C} = -\boldsymbol{H}(\boldsymbol{x}^*)^{-1}$, i.e.

$$q(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{D}{2}}|\boldsymbol{C}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}^*)^{\mathsf{T}}\boldsymbol{C}^{-1}(\boldsymbol{x}-\boldsymbol{x}^*)\right). \tag{2.53}$$

An example of applying Laplace's method to fit a normal approximation to a univariate generalised logistic target is shown in Figure 2.10.

As $q(\boldsymbol{x}^*) \approx p(\boldsymbol{x}^*) = \tilde{p}(\boldsymbol{x}^*)/Z$ we can also form an approximation $\tilde{Z}$ to the normalising constant Z for the target density

$$Z \approx \tilde{Z} = (2\pi)^{\frac{D}{2}}|\boldsymbol{C}|^{\frac{1}{2}}\tilde{p}(\boldsymbol{x}^*). \tag{2.54}$$

To use Laplace's method we need to be able to find a maxima of $\log\tilde{p}$ and evaluate the Hessian at this point. For simple unimodal target densities it may be possible to find the maxima and corresponding Hessian analytically. More generally if the gradient of $\log\tilde{p}$ can be calculated (using for example reverse-mode automatic differentation), then a maxima can be found by performing iterative gradient ascent. The Hessian can then be evaluated at this point using analytic expressions for the second partial derivatives or again by using automatic differentiation (by computing the Jacobian of the gradient of $\log\tilde{p}$).

Though relatively simple to calculate, Laplace's method will often resulting in an approximate density which fits poorly to the target. As it only uses local information about the curvature of the (log) target density at the mode, away from the mode the approximate density can behave very differently from the target density, for instance observe the poor fit to the tails of the target of the example shown in Figure 2.10. For multimodal densities, several different Laplace approximations can be calculated, each likely to at best capture a single mode well. For target densities which are well approximated by a normal distribution, for instance due to asymptotic convergence to normality of a posterior for iid data, Laplace's method can give reasonable results however.

2.2.2 Variational inference

Laplace's method is limited by using information about the target density evaluated at only one point to fit the approximation. An alternative approach is to instead try to fit the approximate density based on minimising a global measure of 'goodness of fit' to the target; this is the strategy employed in *variational inference*.

The naming of variational inference arises from its roots in the *calculus of variations*, which is concerned with *functionals* (loosely a function of a function, often defined by a definite integral) and their derivatives. In particular it is natural to define the measure of the 'goodness of fit' of the approximate density to the target as a functional of the approximate density. The value of this functional is then minimised with respect to the approximate density function.

The most common functional used to define goodness of fit in variational inference is the *Kullback–Leibler* (KL) divergence [62]. The KL divergence in its most general form is defined for a pair of probability measures P and Q on a space X with P absolutely continuous with respect to Q as

$$\mathbb{D}_{\text{KL}}[P \,\|\, Q] = \int_X \log\left(\frac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}P, \tag{2.55}$$

which is read as the KL divergence from P to Q. The KL divergence is always non-negative $\mathbb{D}_{\text{KL}}[P \,\|\, Q] \geq 0$, with equality if and only if $P = Q$ almost everywhere. Intutively the KL divergence gives a measure of how 'close' two measures are[9]. It is not a true distance however as it is asymmetric: in general $\mathbb{D}_{\text{KL}}[P \,\|\, Q] \neq \mathbb{D}_{\text{KL}}[Q \,\|\, P]$.

[9] From an information theory perspective $\mathbb{D}_{\text{KL}}[P \,\|\, Q]$ is typically termed the *relative entropy of P with respect to Q* and measures the expected information loss (in *nats* for base-e logarithms or *bits* for base-2 logarithms) of using Q to model samples from P.

Generally we will work with probability densities rather than underlying probability measures. If p and q are the densities of two probability measures P and Q defined with respect to the same base measure μ on a space X, i.e. $p = \frac{\mathrm{d}P}{\mathrm{d}\mu}$ and $q = \frac{\mathrm{d}Q}{\mathrm{d}\mu}$, then we will denote the KL divergence from P to Q in terms of the densities p and q by $\mathbb{D}^{\mu}_{\mathrm{KL}}[p \,\|\, q] = \mathbb{D}_{\mathrm{KL}}[P \,\|\, Q]$, and from the definition (2.55) we have that

$$\mathbb{D}^{\mu}_{\mathrm{KL}}[p \,\|\, q] = \int_X p(x) \log \frac{p(x)}{q(x)} \,\mathrm{d}\mu(x), \tag{2.56}$$

with absolute continuity of P with respect to Q corresponding to a requirement that $p(x) = 0 \;\forall x \in X : q(x) = 0$. Somewhat loosely, we will refer to $\mathbb{D}^{\mu}_{\mathrm{KL}}[p \,\|\, q]$ as the KL divergence from the (density) p to the (density) q rather than refering to the underlying measures.

When used without further qualification, variational inference is generally intended to mean inference performed by minimising a variational objective corresponding to the KL divergence from an approximate density q to the target density p. More specifically using the decomposition of the target density into an unnormalised density $\tilde{p}$ and normalising constant Z we have that

$$\mathbb{L}[q] = \log Z - \mathbb{D}^{\mu}_{\mathrm{KL}}[q \,\|\, p] = \int_X q(\boldsymbol{x}) \log \frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})} \,\mathrm{d}\mu(\boldsymbol{x}), \tag{2.57}$$

with $\mathbb{L}[q]$ the specific objective usually maximised in variational inference problems, with all terms in the integrand being evaluable pointwise. As $\log Z$ is constant with respect to the approximate density, maximising $\mathbb{L}$ with respect to q is directly equivalent to minimising $\mathbb{D}^{\mu}_{\mathrm{KL}}[q \,\|\, p]$. Due to the non-negativity of the KL divergence we have that the following inequality holds

$$\mathbb{L}[q] \leq \log Z. \tag{2.58}$$

When the target density p corresponds to a posterior $\mathsf{p}_{\mathbf{x}|\mathbf{y}}$ on latent variables $\mathbf{x}$ given observed variables $\mathbf{y}$ and $\tilde{p}$ the corresponding joint density $\mathsf{p}_{\mathbf{x},\mathbf{y}}$, the normalising constant Z is equal to the model evidence term $\mathsf{p}_{\mathbf{x}}$ in Bayes' theorem. As $\mathbb{L}$ is a lower bound on $\log Z$ and so the (log) model evidence, the variational objective $\mathbb{L}$ is therefore sometimes termed the *evidence lower bound* (ELBO) in this context.

Using the KL divergence from the approximate to target density as the variational objective is not the only choice avaialable. One obvious alternative is the reversed form of the KL divergence, $\mathbb{D}^{\mu}_{\mathrm{KL}}[p \,\|\, q]$ from the target density to the approximate density. In general as this form of the divergence involves evaluating an integral with respect to the target density, precisely the intractable computational task we are hoping to find an approximate solution, direct applications of this approach are limited to toy problems were this integral can be solved exactly or efficiently approximated. An approach called *expectation propagation* (EP) [75] however locally optimises an objective closely related to $\mathbb{D}^{\mu}_{\text{KL}}[p \,\|\, q]$.

The KL divergence can be considered as a special case of a broader class of α-divergences. In particular the *Rényi divergence* [27, 94] of order $\alpha > 0, \alpha \neq 1$ between two probability measures P and Q with probability densities $p = \frac{\mathrm{d}P}{\mathrm{d}\mu}$ and $q = \frac{\mathrm{d}Q}{\mathrm{d}\mu}$ on a space X is defined as

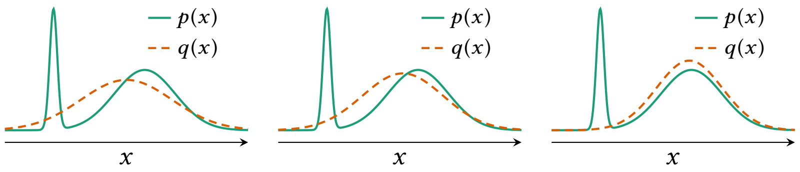
$$\mathbb{D}_{\alpha}[P \,\|\, Q] = \mathbb{D}^{\mu}_{\alpha}[p \,\|\, q] = \frac{1}{\alpha - 1} \log\left(\int_X p(\boldsymbol{x})^{\alpha}\, q(\boldsymbol{x})^{1-\alpha}\, \mathrm{d}\mu(\boldsymbol{x}) \right). \quad (2.59)$$

For $\alpha > 0$, $\mathbb{D}_{\alpha}[P \,\|\, Q]$ is a valid divergence, that is $\mathbb{D}_{\alpha}[P \,\|\, Q] \geq 0$ with equality if and only if $P = Q$ almost everywhere. The definition can also be extended to the cases $\alpha = 1$ and $\alpha = 0$ by considering limits of (2.59). Using L'Hôpital's rule it can be shown that $\lim_{\alpha \to 1} \mathbb{D}_{\alpha}[P \,\|\, Q] = \mathbb{D}_{\text{KL}}[P \,\|\, Q]$. For $\alpha \to 0$, we have that $\mathbb{D}_{\alpha}[P \,\|\, Q] \to -\log P(\text{supp}(Q))$ where $\text{supp}(Q)$ represents the support of the probability measure Q; in this case $\mathbb{D}_{\alpha}[P \,\|\, Q]$ is no longer a valid divergence as it is equal to zero whenever $\text{supp}(P) = \text{supp}(Q)$. It can also be shown that for $\alpha \notin \{0, 1\}$ that $\mathbb{D}_{\alpha}[P \,\|\, Q] = \frac{\alpha}{1-\alpha} \mathbb{D}_{1-\alpha}[Q \,\|\, P]$. This motivates extending the definition in (2.59) for $\alpha < 0$, in which case we have that $\mathbb{D}_{\alpha}[P \,\|\, Q] = \frac{\alpha}{1-\alpha} \mathbb{D}_{1-\alpha}[Q \,\|\, P] \leq 0$ [65].

Analogously to using the decomposition of the target density p in to an unnormalised density $\tilde{p}$ and unknown normaliser Z when defining the previous variational objective in (2.57), it is observed in [65] that a *variational Rényi bound*, $\mathbb{L}_{\alpha}$, can be defined as

$$\mathbb{L}_{\alpha}[q] = \log Z - \mathbb{D}^{\mu}_{\alpha}[q \,\|\, p] = \frac{1}{1 - \alpha} \log \int_X q(\boldsymbol{x}) \left(\frac{\tilde{p}(\boldsymbol{x})}{q(\boldsymbol{x})} \right)^{1-\alpha} \mathrm{d}\mu(\boldsymbol{x}). \quad (2.60)$$

For $\alpha > 0$, we have that $\mathbb{D}^{\mu}_{\alpha}[q \,\|\, p] \geq 0$ and so $\mathbb{L}_{\alpha}$ is a lower bound on the $\log Z$, analogously to the ELBO, and we should maximise $\mathbb{L}_{\alpha}$ with respect to q to minimise $\mathbb{D}^{\mu}_{\alpha}[q \,\|\, p]$. For $\alpha < 0$ we have instead that $\mathbb{D}^{\mu}_{\alpha}[q \,\|\, p] \leq 0$ and so $\mathbb{L}_{\alpha}$ is an upper bound on $\log Z$ and that we should minimise $\mathbb{L}_{\alpha}$ to minimise $\mathbb{D}^{\mu}_{1-\alpha}[p \,\|\, q]$ (note the swapped order of the



(a) $\mathbb{D}^{\lambda}_{\mathrm{KL}}[p \,\|\, q]$ (b) $\mathbb{D}^{\lambda}_{\alpha}[p \,\|\, q]$, $\alpha = \frac{1}{2}$ (c) $\mathbb{D}^{\lambda}_{\mathrm{KL}}[q \,\|\, p]$

Figure 2.11: Comparison of approximate densities fitted under different variational objectives. Each plot shows a bimodal target density $p(x)$ and a normal approximate density $q(x) = \mathcal{N}\left(x \,|\, \mu, \sigma^2\right)$ where μ and σ have been set to values which minimise the variational objective shown in the caption.

density arguments). An equivalent observation of the possibility of upper bounding $\log Z$ is made in [25] with a reparameterised version of (2.60) in terms of $n = 1 - \alpha > 1$.

As generally the family chosen for the approximate density q will not include the target density as a member, the choice of variational objective is important in determining the properties of how q approximates the target density [12]. The standard variational objective corresponding to $\mathbb{D}^{\mu}_{\mathrm{KL}}[q \,\|\, p]$ strongly penalises regions in X where $\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \ll 1$, therefore the approximate densities fitted using this objective tend to be undispersed compared to the target density, and in the case of target densities with multiple separated modes fitted with a unimodal approximate density, the approximate density will tend to fit only one mode well (with fits to the different modes corresponding to different local optima in the objective). Conversely using the reversed KL divergence $\mathbb{D}^{\mu}_{\mathrm{KL}}[p \,\|\, q]$ as the variational objective penalises approximate densities where $\frac{q(\boldsymbol{x})}{p(\boldsymbol{x})} \ll 1$ in regions with significant mass under the target density, therefore the approximate densities fitted using this objective tend to be overdispersed compared to the target density, and in the case of multimodal target densities, the approximate densities will tend to 'cover' multiple modes. Using a variational objective corresponding to a Rényi divergence with $0 < \alpha < 1$, allows interpolating between these two behaviours (with α close to one favouring undispersed approximate densities similar to $\mathbb{D}^{\mu}_{\mathrm{KL}}[q \,\|\, p]$, with the solutions becoming increasingly dispersed as α becomes lower).

Figure 2.11 gives examples of normal approximate densities fitted to a bimodal target with three variational objectives to illustrate the effect of the different objectives on the fitted approximation. In Figure

2.11a the approximate density q was fitted by minimising $\mathbb{D}^{\lambda}_{\text{KL}}[p \,\|\, q]$, the resulting q putting mass on both modes in the target (and significant mass on the region of low density between the two target modes). The approximate density q in Figure 2.11c was instead fitted by minimising $\mathbb{D}^{\lambda}_{\text{KL}}[q \,\|\, p]$, with the result that q concentrates its mass around one of the modes. Finally Figure 2.11b shows an approximate density fitted by minimising the Rényi divergence (2.59) with $\alpha = \frac{1}{2}$ for which $\mathbb{D}^{\lambda}_{\alpha}[p \,\|\, q] = \mathbb{D}^{\lambda}_{\alpha}[q \,\|\, p]$ and which interpolates between the behaviours of the two objectives used in Figures 2.11a and 2.11c. The approximate density here is less dispersed than in the $\mathbb{D}^{\lambda}_{\text{KL}}[p \,\|\, q]$ case, but still places more mass on the minor mode than the $\mathbb{D}^{\lambda}_{\text{KL}}[q \,\|\, p]$ case.

Once the variational objective has been defined, it still remains to choose the family of the approximate density q and optimisation scheme. A very common choice is to use an approximate density in the *mean-field variational family*; this assumes that the variables the target density is defined on can be grouped in to a set of mutually independent vectors $\{\mathbf{x}_i\}_{i \in I}$ and so the approximate density can be factorised as

$$q(\boldsymbol{x}) = \prod_{i \in I} q_i(\boldsymbol{x}_i). \tag{2.61}$$

This assumption can signficantly reduce the computational demands of variational inference and facilitates simple evaluation of the approximate marginal density q_i of each variable group once fitted. However the mutual independence assumption prevents the approximate density q from being able to represent any of the dependencies between the variable groups in the target density. The early development of variational inference was largely based around mean-field family approximations [89, 108], with the naming arising from its origins in *mean-field theory*, used to study the behaviour of systems such as the Ising spin model in statistical physics [86]. Despite the limitations in representational capacity imposed by the independence assumption, because of its computational tractability variational inference using mean-field family approximate densities remains very popular [14].

The mean-field family supports a particularly simple algorithm for optimising the standard variational objective (2.57), *coordinate ascent variational inference* (CAVI) [12, 14]. If we define for each variable group vector $\boldsymbol{x}_i$ a corresponding vector $\boldsymbol{x}_{\setminus i} = [\boldsymbol{x}_j]_{j \in I \setminus i}$ concatenating all the remaining variables, then it can be shown that the optimal factors of a mean-field family approximate density satisfy

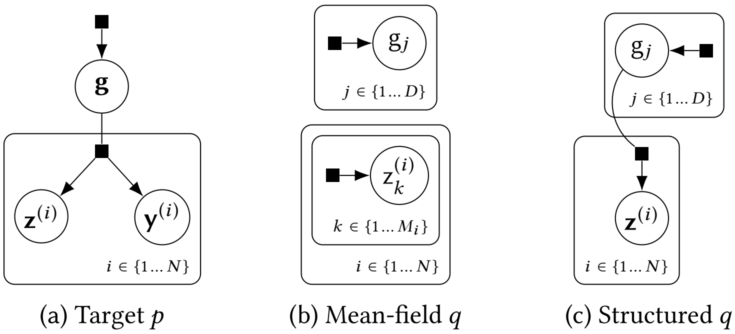
$$q_i(\boldsymbol{x}_i) \propto \exp\left(\int \prod_{j \in I \setminus i} \big(q_j(\boldsymbol{x}_j) \big) \log \tilde{p}(\boldsymbol{x}_i, \boldsymbol{x}_{\setminus i}) \, \mathrm{d}\boldsymbol{x}_{\setminus i} \right). \tag{2.62}$$

The optimal value for each factor is coupled to the values of all of the other factors and so cannot be explicitly solved for even when the integral in (2.62) has an analytic solution. CAVI therefore uses a fixed point iteration approach, sequentially updating each of the factors q_i according to (2.62) given the current values of the remaining factors. This iterative update scheme is guaranteed to eventually converge to a local optimum with all factors satisfying (2.62).

The key computation in CAVI is computing the integral in (2.62) for the updates to each factor. For models with target densities where the conditional densities on each variable group $\boldsymbol{x}_i$ given the remaining variables $\boldsymbol{x}_{\setminus i}$ (termed the *complete conditionals*) are all exponential family densities, an optimal parameteric form for each of the q_i factors can be analytically derived. The optimal factors have a density from the same exponential family as the corresponding complete conditional, with the integral in (2.62) having a closed form solution in this case.

For a model defined by a directed factor graph, a sufficient condition for the complete conditionals to all be exponential family densities is that all factors correspond to exponential family densities and that the factors specifying the (conditional) densities on any parent nodes to a factor are conjugate to the density on the child of the factor (in the sense of Section 1.3.5); such models are termed *conjugate exponential*.

Variational message passing [125], is a CAVI algorithm for performing inference in conjugate exponential models. It exploits factorisation structure in the target density, typically described by a directed graphical model or factor graph, to efficiently update the factors. General purpose implementations are available in software frameworks such as VIBES [13] and Infer.NET [77] which can automatically perform inference given a model specification. The conjugate exponential assumptions can be partially relaxed to also allow deterministic nodes which are multilinear functions of their parents and truncated forms of some exponential family densities which still admit analytic solutions to the factor updates [125].



(a) Target p (b) Mean-field q (c) Structured q

Figure 2.12: (a) Factor graph of a model with global latent variables $\mathbf{g}$, per-datapoint local latent variables $\{\mathbf{z}^{(i)}\}_{i=1}^{N}$ and observed variables $\{\mathbf{y}^{(i)}\}_{i=1}^{N}$. (b) and (c) Factor graphs for mean-field and structured variational approximate densities for model shown in (a).

A more recent alternative to CAVI for mean-field variational inference is *stochastic variational inference* (SVI) [48, 107]. SVI is designed for a common class of models consisting of a set of global latent variables $\mathbf{g}$ plus a set of local latent variables $\{\mathbf{z}^{(i)}\}_{i=1}^{N}$ each associated with one of N observed data points $\{\mathbf{y}^{(i)}\}_{i=1}^{N}$. Each pair of observed and local latent variable $(\mathbf{y}^{(i)}, \mathbf{z}^{(i)})$ are conditionally independent from all the others given the global latent variables; this factorisation structure is visualised in Figure 2.12c. The hierarchical model for the *Observing Dark Worlds* problem encountered earlier for example matches this structure.

CAVI requires a complete pass through all local latent variables for each update to the global latent variables. As the data set size N grows large this can become onerous computationally. Intuitively we might expect that redundancy in the data should mean that a subset of the data points should contain sufficient information to update the approximate density factors for the global variables, particularly early on in the optimisation when far from convergence and so even noisy information can allow significant improvements. The fixed-point iteration of CAVI does not however easily lend itself to exploiting this intuition.

If we instead consider using gradient ascent to maximise the objective, then the gradient of the ELBO objective with respect to the natural parameters of the global variable approximate density factors takes the form of a sum of N terms each dependent on a local latent variable and observation pair $(\mathbf{z}^{(i)}, \mathbf{y}^{(i)})$. We can form an unbiased estimate of this gradient by sampling a subset of M of the local latent variable and observation pairs (commonly $M = 1$). We can then leverage stochastic optimisation methods [96] which are designed precisely to work in this setting, of optimising an objective given a noisy but unbiased estimate of it gradient with respect to parameters.

It is assumed in SVI that the model is conjugate exponential, this meaning the gradients of the variational objective with respect to the natural parameters of the approximate density factors on both local and global latent variables can be computed in closed form. Further in this case of conjugate exponential models, the *natural gradients* [1] with respect to the variational parameters can be efficiently computed; the natural gradient exploits the differential geometry of the natural parameter space (i.e. that it is a Riemannian manifold) by rescaling the standard (Euclidean) gradient by the inverse of a Riemannian metric for the natural parameter manifold. SVI uses stochastic gradient ascent with noisy estimates of the natural parameter natural gradients to allow efficient mean-field variational inference with large datasets.

The methods discussed so far have only applied when using an approximate density in the mean-field family. An alternative is to use a more structured factorisation which reflects some or all of the known dependencies between variables in the target density [3, 49, 109, 110, 112]. These *structured variational inference* approaches use known dependency information such as from a factor graph of the target density, to inform the choice of approximate density factorisation. In general structured variational inference methods will still put some constraints on the factorisation of the approximate density to maintain tractability.

For example *structured stochastic variational inference* [49] applies to the same class of conjugate exponential models as SVI, i.e. with the factorisation structure shown in Figure 2.12c. It extends on SVI by allowing the approximate density to account for dependencies between the global latent variables and local latent variables, assuming a structured factorsiation corresponding to that shown in the factor graph in Figure 2.12c as opposed to the typical mean-field factorisation shown in 2.12b which would be used in standard SVI. This improves on the mean-field approximate density by including the dependencies between the local latent variables and on the global latent variables, but still requires an assumption of independence between the global latent variables.

The variational inference methods considered so far have made the strong assumption that the model being approximated is conjugate exponential. Although the analytic updates to factors made possibly by this assumption offers significant advatantages in terms of the computational tractability and stability of the optimisation of the approximate density (factors), conjugacy is a restrictive assumption which excludes many useful models. For instance the model proposed in the previous chapter for the *Observing Dark Worlds* problem is not conjugate exponential and even seemingly 'simple' models such as a logistic regression model for binary classification with a Gaussian prior on the regression weights breaks conjugacy assumptions.

Various extensions have been proposed for applying mean-field CAVI to non-conjugate exponential models where exact analytic solutions to the factor updates are no longer available. *Non-conjugate variational message passing* [58] describes an extension of the variational message passing framework to models with non-conjugate factors, and provides a concrete algorithm for logistic regression models using model specific bounds on the factor update integrals for which analytic solutions are not available. In [123] an alternative approach with less model specific derivation is proposed. Laplace's method is used to approximate integrals with respect to non-conjugate factors, with a further option of using a Taylor series approximation to the underlying variational objective. It has also been proposed to use quadrature and tractable mixture density approximations to individual factors to approximate solutions to intractable integrals in non-conjugate factor updates [122].

A proposed unifying view of many of the mean-field methods developed for dealing with non-conjugate models, is that they relax the assumption that the approximate factors take the 'non-parameteric' optimal form given by the solution to (2.62), which is derived using a variational approach, and instead assume a fixed parameteric form for some or all of the approximate density factors [102]. This links these methods to other variational inference approaches which assume a fixed parametric form for the whole approximate density, i.e. $q(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, where $f_{\boldsymbol{\theta}}$ is a density of a fixed parametric family with a vector of parameters $\boldsymbol{\theta}$ [7, 40, 56, 85, 93, 95, 105].

Under this parametric assumption, rather than a variational optimisation problem we can now consider the variational objective functional $\mathbb{L}[q]$ as instead a function of the parameters $\ell(\boldsymbol{\theta}) = \mathbb{L}[f_{\boldsymbol{\theta}}]$. Typically the integrals involved in evaluating the parameteric variational objective $\ell(\boldsymbol{\theta})$ (and its gradients) cannot be solved analytically however which seems to leave us with the same problems as encountered when trying to use standard mean-field variational inference approaches with non-conjugate models. By using the sampling methods we will discuss in the next section of this chapter however it is possible estimate these integrals.

Under this parametric assumption, rather than a variational optimisation problem we can now consider the variational objective functional $\mathbb{L}[q]$ as instead a function of the parameters $\ell(\boldsymbol{\theta}) = \mathbb{L}[f_{\boldsymbol{\theta}}]$. For the standard variational objective (2.57) we have that

$$\ell(\boldsymbol{\theta}) = \int_X f_{\boldsymbol{\theta}}(\boldsymbol{x}) \log \tilde{p}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} - \int_X f_{\boldsymbol{\theta}}(\boldsymbol{x}) \log f_{\boldsymbol{\theta}}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} \,\mathrm{d}\boldsymbol{x}. \tag{2.63}$$

By differentiating with respect to $\boldsymbol{\theta}$ and using the identity that for any $f_{\boldsymbol{\theta}}$ which is differentiable with respect to the $\boldsymbol{\theta}$

$$\frac{\partial f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \boldsymbol{\theta}} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) \frac{\partial \log f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \boldsymbol{\theta}} \tag{2.64}$$

we have that

$$\frac{\partial \ell}{\partial \boldsymbol{\theta}} = \int_X f_{\boldsymbol{\theta}}(\boldsymbol{x}) \frac{\partial \log f_{\boldsymbol{\theta}}(\boldsymbol{x})}{\partial \boldsymbol{\theta}} \left(\log\left(\frac{\tilde{p}(\boldsymbol{x})}{f_{\boldsymbol{\theta}}(\boldsymbol{x})} \right) - 1 \right) \mathrm{d}\boldsymbol{x}. \tag{2.65}$$

In [105] $f_{\boldsymbol{\theta}}$ is chosen as an exponential family distribution and $\boldsymbol{\theta}$ specified to be the natural parameters of the density. A linear-regression inspired algorithm w

For instance for the *Observing Dark Worlds* hierarchical model (Figure 1.12) discussed in the previous chapter, we have that the local latent variables corresponding to the halo masses $\mathbf{m}^{(i)}$, core radii $\mathbf{t}^{(i)}$ and centre coordinates $\mathbf{x}^{(i)}, \mathbf{y}^{(i)}$ (for the test set data) for a particular cluster are conditionally independent of the variables for all other clusters given the global variables $\sigma, \mu_\mathrm{m}, \sigma_\mathrm{m}, \mu_\mathrm{t}$ and σ_t. A natural structured factorisation for an approximate density in this case would therefore be that indicated by the factor graph in Figure **??**.

Even with a factorisation chosen for the approximate density, it remains to define the parametric form for each of the approximate factors. In general there is a tradeoff in the choice of approximate density parameterisation between representational power of the resulting approximate density and corresponding ability to represent the target density well, and the tractability of optimising the variational objective.

2.3 HYBRID APPROACHES

3 HAMILTONIAN MONTE CARLO

4 THERMODYNAMIC METHODS

4.1 SIMULATED TEMPERING

4.2 PARALLEL TEMPERING

4.3 TEMPERED TRANSITIONS

4.4 ANNEALED IMPORTANCE SAMPLING

4.5 PATH SAMPLING

4.6 ADIABATIC MONTE CARLO

5 REPARAMETERISATION

Just as there are multiple ways to formulate a computation graph depending on what are used as the intermediate variables and operations, there is flexibility in how we parametrise probabilistic factors in a factor graph. One immediate example of this comes from the above discussion of simulator models. Typically all random number generator routines in a numerical computing library for distributions on real random variables or vectors will be implemented by generating a set of standard uniform random variables using a base pseudo-random number generator and then performing a series of deterministic operations to the uniform variables to produce random variables with the required density. Therefore rather than directly representing a non-uniform directed factor with output x in a factor graph, we could instead choose to *reparametrise* the factor graph in terms of the the uniform random variables $\mathbf{u}$ which are used to generate x, with x now the output of a deterministic factor with input $\mathbf{u}$ corresponding to the deterministic transformation used to produce x with the required density - pictorally ■→ⓧ is transformed to ■→ⓤ—◇→ⓧ.

More generally by applying the change of variables formulae from Section 1.1.4 we can find transformations of random variables such that the transformed random variable has a density of interest. A common motivation we will have for reparameterising a probabilistic model is to 'standardise' the joint density prior to conditioning on any observed values. In particular when performing inference it will often be helpful to parameterise models as far as possible in terms independent unit-variance random variables with unbounded support, for instance $\mathcal{N}(0,1)$, $\mathrm{InvCosh}(0,1)$ or $\mathrm{Logistic}(0,\sqrt{3}/\pi)$. The densities for all three shown for comparison in Figure 5.1. Transforming variables to have unbounded support is usually helpful as its avoids the difficulties of working with constrained spaces. Setting all variables to have unit-variance a-priori helps to normalise the scale of variables, which will often for example simplify choosing step size parameters. Parameterising in terms of independent random variables removes any dependencies prior to

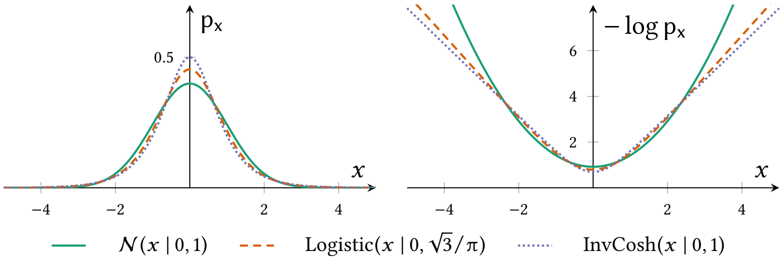


Figure 5.1: Unit variance densities with unbounded support.

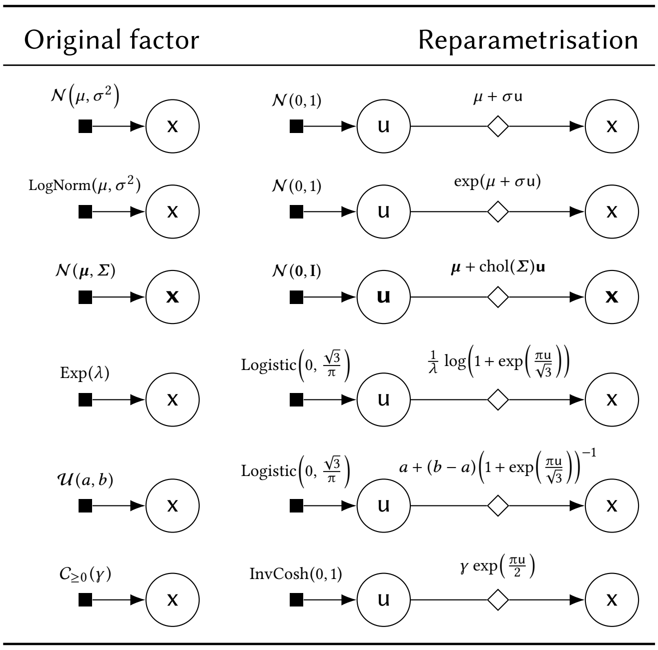
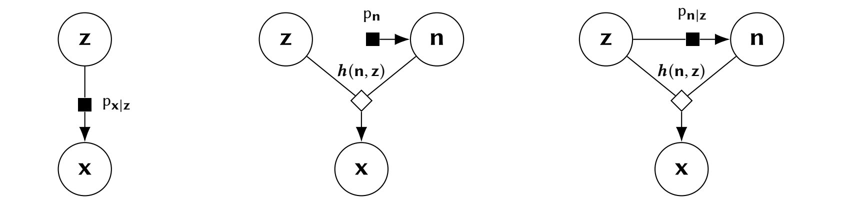

Original factor
Reparametrisation
$\mathcal{N}(\mu, \sigma^2)$
x
$\mathcal{N}(0,1)$
u
$\mu + \sigma \mathrm{u}$
x
$\mathrm{LogNorm}(\mu, \sigma^2)$
x
$\mathcal{N}(0,1)$
u
$\exp(\mu + \sigma \mathrm{u})$
x
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\mathbf{x}$
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{u}$
$\boldsymbol{\mu} + \mathrm{chol}(\boldsymbol{\Sigma})\mathbf{u}$
$\mathbf{x}$
$\mathrm{Exp}(\lambda)$
x
$\mathrm{Logistic}\left(0, \frac{\sqrt{3}}{\pi}\right)$
u
$\frac{1}{\lambda} \log\left(1 + \exp\left(\frac{\pi \mathrm{u}}{\sqrt{3}}\right)\right)$
x
$\mathcal{U}(a, b)$
x
$\mathrm{Logistic}\left(0, \frac{\sqrt{3}}{\pi}\right)$
u
$a + (b - a)\left(1 + \exp\left(\frac{\pi \mathrm{u}}{\sqrt{3}}\right)\right)^{-1}$
x
$\mathcal{C}_{\geq 0}(\gamma)$
x
$\mathrm{InvCosh}(0, 1)$
u
$\gamma \exp\left(\frac{\pi \mathrm{u}}{2}\right)$
x

Table 5.1: Standardisation reparametrisations.

conditioning on observations, with the resulting joint densities typically having geometries which are easier for inference algorithms to handle.


z
$p_{\mathbf{x}|\mathbf{z}}$
x
z
$p_{\mathbf{n}}$
n
$h(\mathbf{n}, \mathbf{z})$
x
z
$p_{\mathbf{n}|\mathbf{z}}$
n
$h(\mathbf{n}, \mathbf{z})$
x

(a) Original factor (b) Fully reparameterised (c) Partially reparameterised

Figure 5.2: Full and partial auxiliary reparameterisation transforms of a directed factor node representing a conditional density $p_{\mathbf{x}|\mathbf{z}}$. In full auxiliary reparameterisation an auxiliary random variable $\mathbf{n}$ is introduced which is (unconditionally) independent of $\mathbf{z}$ such that the output of a deterministic transformation $\mathbf{x} = \boldsymbol{h}(\mathbf{n}, \mathbf{z})$ has the same conditional density given $\mathbf{z}$ as the original factor output. In partial auxiliary reparameterisation the auxiliary random variable $\mathbf{n}$ is instead conditionally dependent on $\mathbf{z}$.

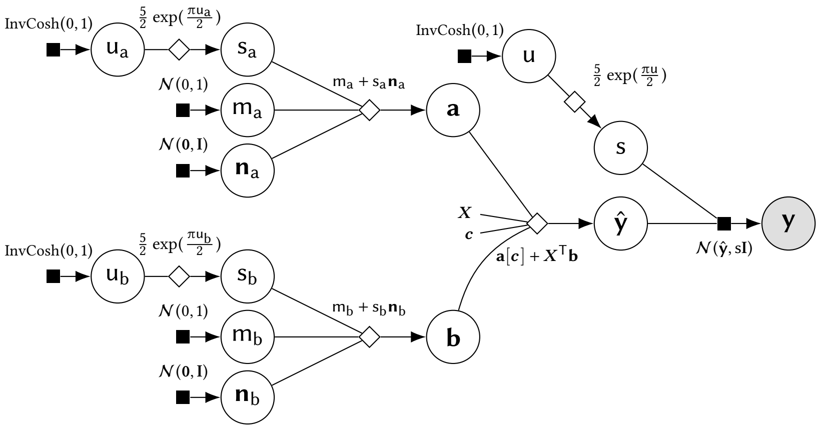

InvCosh(0, 1)
u_a
$\frac{5}{2}\exp(\frac{\pi u_a}{2})$
s_a
$\mathcal{N}(0, 1)$
m_a
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{n}_a$
$m_a + s_a\mathbf{n}_a$
a
InvCosh(0, 1)
u
$\frac{5}{2}\exp(\frac{\pi u}{2})$
s
X
$\boldsymbol{c}$
$\mathbf{a}[\boldsymbol{c}] + X^{\mathsf{T}}\mathbf{b}$
$\hat{\mathbf{y}}$
$\mathcal{N}(\hat{\mathbf{y}}, s\mathbf{I})$
y
InvCosh(0, 1)
u_b
$\frac{5}{2}\exp(\frac{\pi u_b}{2})$
s_b
$\mathcal{N}(0, 1)$
m_b
$\mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{n}_b$
$m_b + s_b\mathbf{n}_b$
b

Figure 5.3: Reparametrised hierarchical linear regression model factor graph.

BIBLIOGRAPHY

[1] Shun-Ichi Amari. ‘Differential geometry of curved exponential families-curvatures and information loss’. In: *The Annals of Statistics* (1982), pp. 357–385.

[2] IEEE Standards Association. ‘Standard for Floating-Point Arithmetic’. In: *IEEE 754-2008* (2008).

[3] David Barber and Wim Wiegerinck. ‘Tractable variational structures for approximating graphical models’. In: *Advances in Neural Information Processing Systems*. 1999, pp. 183–189.

[4] Alessandro Barp, Francois-Xavier Briol, Anthony D Kennedy and Mark Girolami. ‘Geometry and Dynamics for Markov Chain Monte Carlo’. In: *arXiv preprint arXiv:1705.02891* (2017).

[5] Matthias Bartelmann and Peter Schneider. ‘Weak gravitational lensing’. In: *Physics Reports* 340.4 (2001), pp. 291–472.

[6] Friedrich L Bauer. ‘Computational graphs and rounding error’. In: *SIAM Journal on Numerical Analysis* 11.1 (1974), pp. 87–96.

[7] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul and Jeffrey Mark Siskind. ‘Automatic differentiation in machine learning: a survey’. In: *arXiv preprint arXiv:1502.05767* (2015).

[8] L. M. Beda, L. N. Korolev, N. V. Sukkikh and T. S. Frolova. *Programs for automatic differentiation for the machine BESM*. Technical Report. (In Russian). Moscow, USSR: Institute for Precise Mechanics and Computation Techniques, Academy of Science, 1959.

[9] Michael Betancourt. ‘A Conceptual Introduction to Hamiltonian Monte Carlo’. In: *arXiv preprint arXiv:1701.02434* (2017).

[10] Joris Bierkens. ‘Non-reversible Metropolis–Hastings’. In: *Statistics and Computing* 26.6 (2016), pp. 1213–1228.

[11] George D Birkhoff. ‘Proof of the ergodic theorem’. In: *Proceedings of the National Academy of Sciences* 17.12 (1931), pp. 656–660.

[12] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. ISBN: 9780387310732.

[13] Christopher M Bishop, David Spiegelhalter and John Winn. 'VIBES: A variational inference engine for Bayesian networks'. In: *Advances in Neural Information Processing Systems*. Vol. 15. 2002, pp. 777–784.

[14] David M Blei, Alp Kucukelbir and Jon D McAuliffe. 'Variational inference: A review for statisticians'. In: *Journal of the American Statistical Association* just-accepted (2017).

[15] George EP Box. 'Sampling and Bayes' inference in scientific modelling and robustness'. In: *Journal of the Royal Statistical Society. Series A (General)* (1980), pp. 383–430.

[16] George EP Box, Mervin E Muller et al. 'A note on the generation of random normal deviates'. In: *The Annals of Mathematical Statistics* 29.2 (1958), pp. 610–611.

[17] Wray L Buntine. 'Operations for learning with graphical models'. In: *Journal of artificial intelligence research* (1994).

[18] Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li and Allen Riddell. 'Stan: A probabilistic programming language'. In: *Journal of Statistical Software* (2016).

[19] KS Chan. 'Asymptotic behavior of the Gibbs sampler'. In: *Journal of the American Statistical Association* 88.421 (1993), pp. 320–326.

[20] Kung Sik Chan and Charles J Geyer. 'Discussion: Markov chains for exploring posterior distributions'. In: *The Annals of Statistics* 22.4 (1994), pp. 1747–1758.

[21] Richard T Cox. 'Probability, frequency and reasonable expectation'. In: *American Journal of Physics* 14.1 (1946), pp. 1–13. URL: http://dx.doi.org/10.1119/1.1990764.

[22] Richard T Cox. 'The algebra of probable inference'. In: *American Journal of Physics* 31.1 (1963), pp. 66–67. URL: http://dx.doi.org/10.1119/1.1969248.

[23] Philip J. Davis and Philip Rabinowitz. *Numerical Integration*. Blaisdell Publishing Company, 1967.

[24] Persi Diaconis, Susan Holmes and Radford M Neal. ‘Analysis of a nonreversible Markov chain sampler’. In: *Annals of Applied Probability* (2000), pp. 726–752.

[25] Adji B Dieng, Dustin Tran, Rajesh Ranganath, John Paisley and David M Blei. ‘The χ-Divergence for Approximate Inference’. In: *arXiv preprint arXiv:1611.00328* (2016).

[26] Peter J Diggle and Richard J Gratton. ‘Monte Carlo methods of inference for implicit statistical models’. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1984), pp. 193–227.

[27] Tim van Erven and Peter Harremos. ‘Rényi divergence and Kullback-Leibler divergence’. In: *IEEE Transactions on Information Theory* 60.7 (2014), pp. 3797–3820.

[28] Herbert Federer. *Geometric measure theory.* Springer, 1969.

[29] Bruno de Finetti. ‘Foresight: its logical laws, its subjective sources’. In: *Studies in Subjective Probability.* Ed. by H. E. Kyburg. English translation of original 1937 French article *La Prévision: ses lois logiques, ses sources subjectives.* Springer, 1992, pp. 134–174.

[30] Brendan J Frey. ‘Extending factor graphs so as to unify directed and undirected graphical models’. In: *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence.* Morgan Kaufmann Publishers Inc. 2002, pp. 257–264. URL: https://arxiv.org/abs/1212.2486.

[31] Brendan J Frey, Frank R Kschischang, Hans-Andrea Loeliger and Niclas Wiberg. ‘Factor graphs and algorithms’. In: *Proceedings of the 35th Annual Allerton Conference on Communication Control and Computing.* 1997.

[32] Alan E Gelfand and Adrian FM Smith. ‘Sampling-based approaches to calculating marginal densities’. In: *Journal of the American Statistical Association* (1990).

[33] Andrew Gelman, Walter R Gilks and Gareth O Roberts. ‘Weak convergence and optimal scaling of random walk Metropolis algorithms’. In: *The annals of applied probability* 7.1 (1997), pp. 110–120.

[34] Andrew Gelman and Cosma Rohilla Shalizi. ‘Philosophy and the practice of Bayesian statistics’. In: *British Journal of Mathematical and Statistical Psychology* 66.1 (2013), pp. 8–38.

[35] Stuart Geman and Donald Geman. 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images'. In: *IEEE Transactions on pattern analysis and machine intelligence* (1984).

[36] Charles J Geyer. *Markov chain Monte Carlo lecture notes (Statistics 8931)*. Tech. rep. University of Minnesota, 1998.

[37] Charles J Geyer. 'The Metropolis-Hastings-Green Algorithm'. 2003. URL: http://www.stat.umn.edu/geyer/f05/8931/bmhg.pdf.

[38] Wally R Gilks, Andrew Thomas and David J Spiegelhalter. 'A language and program for complex Bayesian modelling'. In: *The Statistician* (1994), pp. 169–177.

[39] Walter R Gilks and Pascal Wild. 'Adaptive rejection sampling for Gibbs sampling'. In: *Applied Statistics* (1992), pp. 337–348.

[40] Alex Graves. 'Practical Variational Inference for Neural Networks'. In: *Advances in Neural Information Processing Systems 24*. 2011, pp. 2348–2356. URL: http://papers.nips.cc/paper/4329-practical-variational-inference-for-neural-networks.pdf.

[41] Peter J Green. 'Reversible jump Markov chain Monte Carlo computation and Bayesian model determination'. In: *Biometrika* (1995), pp. 711–732.

[42] J. E. Gubernatis. 'Marshall Rosenbluth and the Metropolis algorithm'. In: *Physics of Plasmas* 12.5 (2005), p. 057303. URL: http://dx.doi.org/10.1063/1.1887186.

[43] Heikki Haario, Eero Saksman and Johanna Tamminen. 'An adaptive Metropolis algorithm'. In: *Bernoulli* (2001), pp. 223–242.

[44] Theodore E Harris. 'The existence of stationary measures for certain Markov processes'. In: *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 2. 1956, pp. 113–124.

[45] David Harvey, Thomas D Kitching, Joyce Noah-Vanhoucke, Ben Hamner, Tim Salimans and AM Pires. 'Observing Dark Worlds: A crowdsourcing experiment for dark matter mapping'. In: *Astronomy and Computing* 5 (2014), pp. 35–44.

[46] W Keith Hastings. 'Monte Carlo sampling methods using Markov chains and their applications'. In: *Biometrika* (1970).

[47] Bryan D He, Christopher M De Sa, Ioannis Mitliagkas and Christopher Ré. 'Scan Order in Gibbs Sampling: Models in Which it Matters and Bounds on How Much'. In: *Advances in Neural Information Processing Systems*. 2016, pp. 1–9.

[48] Matthew D Hoffman, David M Blei, Chong Wang and John William Paisley. 'Stochastic variational inference.' In: *Journal of Machine Learning Research* (2013).

[49] Matthew Hoffman and David Blei. 'Structured stochastic variational inference'. In: *Artificial Intelligence and Statistics*. 2015, pp. 361–369.

[50] Akihisa Ichiki and Masayuki Ohzeki. 'Violation of detailed balance accelerates relaxation'. In: *Physical Review E* 88.2 (2013), p. 020101.

[51] Eric Jullo, Jean-Paul Kneib, Marceau Limousin, Ardis Eliasdottir, PJ Marshall and Tomas Verdugo. 'A Bayesian approach to strong lensing modelling of galaxy clusters'. In: *New Journal of Physics* 9.12 (2007), p. 447. URL: https://arxiv.org/abs/0706.0048.

[52] Kaggle. *Observing Dark Worlds*. https://www.kaggle.com/c/DarkWorlds. 2012.

[53] Herman Kahn and Theodore E Harris. 'Estimation of particle transmission by random sampling'. In: *National Bureau of Standards applied mathematics series* 12 (1951), pp. 27–30.

[54] Rudolph Emil Kalman. 'A new approach to linear filtering and prediction problems'. In: *Journal of Basic Engineering* 82.1 (1960), pp. 35–45.

[55] Ross Kindermann and Laurie Snell. *Markov random fields and their applications*. American Mathematical Society, 1980.

[56] Diederik P Kingma and Max Welling. 'Auto-Encoding Variational Bayes'. In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*. 2014. 2013.

[57] P.E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Applications of Mathematics. Springer-Verlag, 1992. ISBN: 9783540540625.

[58] David A Knowles and Tom Minka. 'Non-conjugate variational message passing for multinomial and binary regression'. In: *Advances in Neural Information Processing Systems*. 2011, pp. 1701–1709.

[59] Andreĭ Nikolaevich Kolmogorov. *Foundations of the Theory of Probability.* Ed. by Nathan Morrison. 2nd English Edition. English translation of original 1933 German monograph, *Grundbegriffe der Wahrscheinlichkeitrechnung*. Chelsea Publishing Company, 1956. URL: https://pdfs.semanticscholar.org/c3e1/51f71168a5f348bdebfde11752ca603fa6d0.pdf.

[60] Augustine Kong. *A note on importance sampling using standardized weights.* Technical Report 348. Department of Statistics, University of Chicago, 1992.

[61] Dirk P. Kroese, Thomas Taimre and Zdravko I. Botev. 'Variance Reduction'. In: *Handbook of Monte Carlo Methods.* John Wiley & Sons, Inc., 2011, pp. 347–380. ISBN: 9781118014967. DOI: 10.1002/9781118014967.ch9. URL: http://dx.doi.org/10.1002/9781118014967.ch9.

[62] Solomon Kullback and Richard A Leibler. 'On information and sufficiency'. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[63] Steffen L Lauritzen and David J Spiegelhalter. 'Local computations with probabilities on graphical structures and their application to expert systems'. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1988), pp. 157–224. DOI: 10.2307/2345762.

[64] Derrick H Lehmer. 'Mathematical methods in large-scale computing units'. In: *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery (1949).* 1951, pp. 141–146.

[65] Yingzhen Li and Richard E Turner. 'Rényi divergence variational inference'. In: *Advances in Neural Information Processing Systems.* 2016.

[66] David JC MacKay. *Information theory, inference and learning algorithms.* Cambridge University Press, 2003.

[67] George Marsaglia. 'Random numbers fall mainly in the planes'. In: *Proceedings of the National Academy of Sciences* 61.1 (1968), pp. 25–28.

[68] George Marsaglia, Wai Wan Tsang et al. 'The Ziggurat Method for Generating Random Variables'. In: *Journal of Statistical Software* 5.i08 (2000).

[69] Phillip James Marshall, Michael Paul Hobson and Anže Slosar. ‘Bayesian joint analysis of cluster weak lensing and Sunyaev–Zel’dovich effect data’. In: *Monthly Notices of the Royal Astronomical Society* 346.2 (2003), pp. 489–500.

[70] Richard Massey, Thomas Kitching and Johan Richard. ‘The dark matter of gravitational lensing’. In: *Reports on Progress in Physics* 73.8 (2010), p. 086901.

[71] Makoto Matsumoto and Takuji Nishimura. ‘Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator’. In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 8.1 (1998), pp. 3–30.

[72] Kerrie L Mengersen and Richard L Tweedie. ‘Rates of convergence of the Hastings and Metropolis algorithms’. In: *The annals of Statistics* 24.1 (1996), pp. 101–121.

[73] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller and Edward Teller. ‘Equation of state calculations by fast computing machines’. In: *The Journal of Chemical Physics* (1953).

[74] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 1993.

[75] Thomas P Minka. ‘Expectation propagation for approximate Bayesian inference’. In: *Proceedings of the Seventeenth conference on Uncertainty in Artificial Intelligence* (2001).

[76] Tom Minka and John Winn. ‘Gates: a graphical notation for mixture models’. In: *Advances in Neural Information Processing Systems*. 2009, pp. 1073–1080.

[77] Tom Minka, John Winn, John Guiver, David Knowles, John Bronksill, Sam Webster and Yordan Zakyov. *Infer.NET*. 2014.

[78] Antonietta Mira and Charles J Geyer. ‘On non-reversible Markov chains’. In: *Monte Carlo Methods, Fields Institute/AMS* (2000), pp. 95–110.

[79] Iain Murray. *A Bayesian approach to Observing Dark Worlds*. http://homepages.inf.ed.ac.uk/imurray2/pub/. 2012.

[80] Radford M Neal. ‘Improving asymptotic variance of MCMC estimators: Non-reversible chains are better’. In: *arXiv preprint math/0407281* (2004).

[81] John von Neumann. ‘Various techniques used in connection with random digits’. In: *National Bureau of Standards applied mathematics series* 3 (1951), pp. 36–38.

[82] John F Nolan. ‘Analytical differentiation on a digital computer’. PhD thesis. Massachusetts Institute of Technology, 1953.

[83] Sheehan Olver and Alex Townsend. ‘Fast inverse transform sampling in one and two dimensions’. In: *arXiv preprint arXiv:1307.1223* (2013).

[84] Art B. Owen. ‘Importance sampling’. In: *Monte Carlo theory, methods and examples*. 2013. URL: http://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf.

[85] John W Paisley, David M Blei and Michael I Jordan. ‘Variational Bayesian Inference with Stochastic Search’. In: *Proceedings of the 29th International Conference on Machine Learning*. 2012, pp. 1367–1374.

[86] G. Parisi. *Statistical Field Theory*. Advanced book classics. Avalon Publishing, 1998. ISBN: 9780738200514. URL: https://books.google.co.uk/books?id=y0-8xQOw6FcC.

[87] Judea Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann, 1988.

[88] Peter H Peskun. ‘Optimum Monte-Carlo sampling using Markov chains’. In: *Biometrika* 60.3 (1973), pp. 607–612.

[89] Carsten Peterson and James R Anderson. ‘A mean field theory learning algorithm for neural networks’. In: *Complex systems* (1987).

[90] Martyn Plummer et al. ‘JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling’. In: *Proceedings of the 3rd international workshop on distributed statistical computing*. Vol. 124. Vienna. 2003, p. 125.

[91] Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines. ‘CODA: Convergence Diagnosis and Output Analysis for MCMC’. In: *R News* 6.1 (2006), pp. 7–11. URL: https://journal.r-project.org/archive/.

[92] Adrian E Raftery and Steven Lewis. *How many iterations in the Gibbs sampler?* Tech. rep. Washington University, Seattle, Department of Statistics, 1991.

[93] Rajesh Ranganath, Sean Gerrish and David Blei. ‘Black Box Variational Inference’. In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. 2014.

[94] Alfréd Rényi. ‘On measures of entropy and information’. In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. 1961, pp. 547–561.

[95] Danilo Jimenez Rezende, Shakir Mohamed and Daan Wierstra. ‘Stochastic Backpropagation and Approximate Inference in Deep Generative Models’. In: *Proceedings of The 31st International Conference on Machine Learning*. 2014, pp. 1278–1286.

[96] Herbert Robbins and Sutton Monro. ‘A stochastic approximation method’. In: *The Annals of Mathematical Statistics* (1951), pp. 400–407.

[97] Christian Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*. Springer Science & Business Media, 2007.

[98] Gareth O Roberts and Jeffrey S Rosenthal. ‘Optimal scaling for various Metropolis-Hastings algorithms’. In: *Statistical science* 16.4 (2001), pp. 351–367.

[99] Gareth O Roberts and Jeffrey S Rosenthal. ‘General state space Markov chains and MCMC algorithms’. In: *Probability Surveys* 1 (2004), pp. 20–71.

[100] Gareth O Roberts and Adrian FM Smith. ‘Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms’. In: *Stochastic processes and their applications* 49.2 (1994), pp. 207–216.

[101] Gareth O Roberts and Richard L Tweedie. ‘Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms’. In: *Biometrika* (1996), pp. 95–110.

[102] David Rohde and Matt P Wand. ‘Semiparametric mean field variational Bayes: General principles and numerical issues’. In: *Journal of Machine Learning Research* 17.172 (2016), pp. 1–47.

[103] Jeffrey S Rosenthal et al. ‘Optimal proposal distributions and adaptive MCMC’. In: *Handbook of Markov Chain Monte Carlo* (2011), pp. 93–112.

[104] Tim Salimans. *Observing Dark Worlds*. `http://timsalimans.com/observing-dark-worlds/`. 2012.

[105] Tim Salimans, David A Knowles et al. ‘Fixed-form variational posterior approximation through stochastic linear regression’. In: *Bayesian Analysis* 8.4 (2013), pp. 837–882.

[106] John Salvatier, Thomas V Wiecki and Christopher Fonnesbeck. ‘Probabilistic programming in Python using PyMC3’. In: *PeerJ Computer Science* (2016).

[107] Masa-Aki Sato. ‘Online model selection based on the variational Bayes’. In: *Neural Computation* 13.7 (2001), pp. 1649–1681.

[108] Lawrence K Saul, Tommi Jaakkola and Michael I Jordan. ‘Mean field theory for sigmoid belief networks’. In: *Journal of Artificial Intelligence Research* (1996).

[109] Lawrence K Saul and Michael I Jordan. ‘Exploiting tractable substructures in intractable networks’. In: *Advances in Neural Information Processing Systems*. 1996, pp. 486–492.

[110] Rishit Sheth and Roni Khardon. ‘Monte Carlo Structured SVI for Non-Conjugate Models’. In: *arXiv preprint arXiv:1612.03957* (2016).

[111] Bert Speelpenning. ‘Compiling Fast Partial Derivatives of Functions Given by Algorithms’. PhD thesis. University of Illinois at Urbana-Champaign, 1980.

[112] Amos J Storkey. ‘Dynamic trees: A structured variational method giving efficient propagation rules’. In: *Proceedings of the Sixteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc. 2000, pp. 566–573.

[113] R. L. Stratonovich. ‘Conditional Markov Processes’. In: *Theory of Probability & Its Applications* 5.2 (1960), pp. 156–178. DOI: 10.1137/1105015. URL: http://dx.doi.org/10.1137/1105015.

[114] Yi Sun, Jürgen Schmidhuber and Faustino J Gomez. ‘Improving the asymptotic performance of Markov chain Monte-Carlo by inserting vortices’. In: *Advances in Neural Information Processing Systems*. 2010, pp. 2235–2243.

[115] Hidemaro Suwa and Synge Todo. ‘Markov chain Monte Carlo method without detailed balance’. In: *Physical review letters* 105.12 (2010), p. 120603.

[116] Theano Development Team et al. ‘Theano: A Python framework for fast computation of mathematical expressions’. In: *arXiv e-prints* abs/1605.02688 (May 2016). URL: http://arxiv.org/abs/1605.02688.

[117] Alexander Terenin and David Draper. ‘Cox’s Theorem and the Jaynesian Interpretation of Probability’. arXiv preprint. 2015. URL: https://arxiv.org/abs/1507.06597v2.

[118] Madeleine B Thompson. ‘A comparison of methods for computing autocorrelation time’. In: *arXiv preprint arXiv:1011.0175* (2010).

[119] Luke Tierney. ‘Markov chains for exploring posterior distributions’. In: *The Annals of Statistics* (1994), pp. 1701–1728.

[120] Konstantin S Turitsyn, Michael Chertkov and Marija Vucelja. ‘Irreversible Monte Carlo algorithms for efficient sampling’. In: *Physica D: Nonlinear Phenomena* 240.4 (2011), pp. 410–414.

[121] Stanislaw Ulam and Nicholas Metropolis. ‘The Monte Carlo method’. In: *Journal of the American Statistical Association* 44.247 (1949), pp. 335–341.

[122] Matthew P Wand, John T Ormerod, Simone A Padoan, Rudolf Fuhrwirth et al. ‘Mean field variational Bayes for elaborate distributions’. In: *Bayesian Analysis* 6.4 (2011), pp. 847–900.

[123] Chong Wang and David M Blei. ‘Variational inference in nonconjugate models’. In: *Journal of Machine Learning Research* 14.Apr (2013), pp. 1005–1031.

[124] Robert Edwin Wengert. ‘A simple automatic derivative evaluation program’. In: *Communications of the ACM* 7.8 (1964), pp. 463–464.

[125] John Winn and Christopher M Bishop. ‘Variational message passing’. In: *Journal of Machine Learning Research* 6.Apr (2005), pp. 661–694.