DrSMC: A Sequential Monte Carlo Sampler for Deterministic Relationships on Continuous Random Variables

by

Neil Spencer

B.Sc. with Honours in Mathematics and Statistics, Acadia University, 2013

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

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Statistics)

The University of British Columbia
(Vancouver)

August 2015

Abstract

Computing posterior distributions over variables linked by deterministic constraints is a recurrent problem in Bayesian analysis. Such problems can arise due to censoring, identifiability issues, or other considerations. It is well-known that standard implementations of Monte Carlo inference strategies break down in the presence of these deterministic relationships. Although several alternative Monte Carlo approaches have been recently developed, few are applicable to deterministic relationships on continuous random variables. In this thesis, I propose Deterministic relationship Sequential Monte Carlo (DrSMC), a new Monte Carlo method for continuous variables possessing deterministic constraints. My exposition focuses on developing a DrSMC algorithm for computing the posterior distribution of a continuous random vector given its sum. I derive optimal settings for this algorithm and compare its performance to that of alternative approaches in the literature.

Preface

This dissertation is original, unpublished, independent work by the author, Neil Spencer.

Table of Contents

List of Tables

List of Figures

Acknowledgments

Throughout my time at UBC, I have had the pleasure of being surrounded by many talented friends and mentors. Not only did their support and guidance make this thesis possible, it transformed a daunting and difficult task into a delightful and rewarding experience.

First and foremost, I would like to express my gratitude to my thesis supervisor, Alexandre Bouchard-Côté. Thank you for your confidence, cheerfulness, insightfulness, and for having the patience to tolerate my dabbling in several projects before picking a thesis topic. In particular, thank you for introducing me to and leading such an engaging reading group. To Vincent, Seong, Camila, Crystal, Sean, Sohrab, Reza, Creagh, Elena, Henry, and the rest of the reading group, thank you for your effort and enthusiasm. Much of the work in this thesis was inspired by group presentations and discussions.

Thank you very much to those academics outside of UBC whose correspondences helped make this thesis possible: Shirin Golchi, Liam Paninski, Arnaud Doucet, Michael Betancourt, and Lei Li. Special thanks to Dave Campbell at Simon Fraser, whose helpful comments, encouragement, and patience were very helpful in the writing of this thesis. Thanks to Harry Joe for his insights regarding multivariate normal distributions. Thank you to Pritam Ranjan for getting me into statistics, and for his continued support and encouragement over the years.

Many thanks to my officemates for the countless vibrant discussions in the past two years. We had our share of good times (nights out, suppers in the village, spirited debates), and difficult times (conference deadlines, the "cold wars"). Thanks to Sohrab for his kindness and conscientiousness, Daniel for his enthusiasm and many compliments, Vivian for her pleasant conversations, Andres (honorary of-

ficemate and great excursion planner) for the many long walks (and for making the term “the doctor” stick), and Jack for his friendliness and go-karting skills. Thanks to Creagh for his collaborations, for being a fellow office night-owl, and for his clever jokes. Thanks to Sean for being my collaborator from day one, for his continuous help with technology, for having the patience to put up with my “intuition” and “tastelessness”, and for the Paris grand tour.

Thank you to the rest of my cohort, Chiara, Huiting, Vanny, Jinyuan, Ken, and Eric for their camaraderie. Special thanks to Danny for being my first friend in Vancouver (I think we single-handedly kept the Chinese food places open in our first year). Thank you to Tyler and Vincenzo for hosting delightful parties. Thank you to the rest of the faculty and staff of the Department of Statistics for creating such a comfortable learning environment. Thank you to NSERC for partially funding my research.

Thank you to my dear friends Ben Callaghan and Margaux Ross. You guys provided a delightful escape when the academic life got overwhelming. Ben, I already miss taco/sloppy Joe nights. Thank you to the rest of AVFR for a great frisbee season. Finally, thank you to my family for your continued support and encouragement. This is directed at both my Nova Scotia family: my parents, siblings, grandparents and extended relatives, as well my surrogate Vancouver family: the Ross clan, including Cooper.

Chapter 1

Introduction

Monte Carlo methods provide a rich class of probabilistic inference tools for problems arising in statistics and machine learning. Methods such as Importance Sampling ([21]), Sequential Monte Carlo (SMC), and Markov Chain Monte Carlo (MCMC) ([21]) have fuelled significant advancements in the realm of statistical computation. This is especially true in the realm of Bayesian statistics; Monte Carlo methods are standard for approximating posterior distributions for complex models. This thesis introduces Deterministic Relationship Sequential Monte Carlo (DrSMC), a Monte Carlo method for continuous random variables which are deterministically related.

Deterministic relationships arise in a variety of modelling situations, such as linkage analysis [14], transportation [16], and medical diagnosis [27]. An instance where determinism arises naturally is described by [20], in which a mobile operator wants to simulate the lengths of a sequence of k individual phone calls given that their aggregated length is s. In other cases, deterministic relationships are artificially introduced for practical reasons. For example, when inferring the rate matrix of a continuous time Markov chain, it is common to restrict the space of rate matrices to those in which the expected number of jumps is one per unit of time in order to maintain model identifiability (e.g. [13]).

It is well-known that standard implementations of Monte Carlo methods are not directly applicable in the presence of deterministic relationships ([5, 31]). In such situations, approximation bounds become arbitrarily large and MCMC mixing

rates become arbitrarily slow. To avoid this, [5] recommend transforming the network to remove the deterministic dependencies. However, these transformations can render a network intractably large and be computationally infeasible to carry out [20]. For this reason, there has been a multitude of recent work proposing alternative inference procedures to deal with deterministic relationships. Almost all of this work focuses on discrete variables (e.g. [12, 15, 18]) or other complicated combinatorial variables (such as graphs [3], [30]). The literature for the case of continuous variables is much more sparse, with the applicability of the proposed methods being limited in scope (e.g. deterministic sums with independent and easily sampled marginals [20], unit vectors and the unit simplex [29]).

To address the deficiency in the literature, I introduce DrSMC, a new SMC sampler [9] whose scope consists of sampling problems in which the deterministic relationship can be expressed as $f(X) = s$, where, $X = (X_1, \ldots, X_k)$ is a vector of continuous random variables, f is a continuous almost everywhere function of X, and s is a deterministic value in the range of f. DrSMC involves a gradual enforcement of the constraint $f(X) = s$ using Markov Kernels to move through a series of intermediate distributions in a sequential sampling procedure.

In high-dimensional problems in which f is differentiable, it is useful to employ Hamiltonian Monte Carlo (HMC [22]) when developing the Markov Kernel proposals. HMC's use of the gradient information avoids the dimensionality problems associated with random walks, thereby expediting the rate at which proposed moves satisfy the deterministic relationship.

Despite being developed independently, the general DrSMC algorithm shares many similarities with a special instance of a recently proposed family of samplers called Sequentially Constrained Monte Carlo (SCMC [17]) samplers. Deterministic relationships fall within a broad class of problems to which SCMC is applicable. However, a key difference between DrSMC and SCMC is the functional form used to gradually enforce the constraint. A consequence of this is that, unlike DrSMC, SCMC is unable to utilize HMC kernels. I further elaborate on the differences between DrSMC and SCMC for deterministic problems in Chapter 3.

The remainder of this thesis is organized as follows. Chapter 2 provides relevant background information on importance sampling, SMC, SMC samplers, and Hamiltonian Monte Carlo. Section 3 introduces the general DrSMC procedure,

discusses implementation using both Metropolis Hastings (MH) and HMC kernels. Emphasis is placed a specialized proposal kernel which facilitates tuning DrSMC for deterministic sums. Chapter 4 illustrates DrSMC for deterministic sums ona synthetic problem, comparing its performance to that of DYSC [20] and SCMC [17]. Chapter 5 makes some concluding remarks. The appendix contains some mathematical details that are too lengthy for the main text.

Chapter 2

Background

Deterministic relationship Sequential Monte Carlo algorithm (DrSMC) falls within a broad class of sampling algorithms referred to as Sequential Monte Carlo samplers ([9] SMCS). As a result, exposition and correctness of this algorithm relies heavily on results and insights from the SMCS framework. Section 2.1 reviews importance sampling (IS), sequential importance sampling (SIS), and sequential Monte Carlo (SMC), the foundations of the SMCS framework. Section 2.2 presents the SMCS framework, and highlights its connections with SMC.

Following the discussion of SMCS, Section 2.3 provides a brief review of Hamiltonian Monte Carlo (HMC), a Markov Chain Monte Carlo strategy which can be used to develop effective DrSMC kernels. For ease of presentation, this chapter shall follow [9] and assume any probability measure π admits a density $\pi(x)$ (with a slight abuse of notation) with respected to a σ-finite dominating measure denoted dx. Throughout this thesis, $y_{\ell_1:\ell_2}$ is used to denote the sequence of variables $y_{\ell_1}, y_{\ell+1}, \ldots, y_{\ell_2}$.

2.1 Importance Sampling and Sequential Monte Carlo

The objective of Monte Carlo methods is to approximate a target probability measure π on a measurable space $\{E, \mathscr{E}\}$ with a set of points $x_1, \ldots, x_N \in E$. This is useful for approximating properties of π, namely the expectation $\mathbb{E}_\pi(\psi)$ of a function ψ on E, (e.g. $\psi(x) = x$ for first moment). In simple cases where π can

be sampled from directly, this value can be estimated by sampling a set of points $x_1, \ldots, x_N$ independently from π and invoking the strong law of large numbers. That is, if $\mathbb{E}_\pi(|\psi(x)|) < \infty$,

$$\frac{\sum_{i=1}^{N} \psi(x_N)}{N} \to \int_E \psi(x)\pi(x)dx,$$

where $\to$ denotes almost-sure convergence. However, when π represents a multivariate distribution with complicated dependencies (such as deterministic relationships), it typically cannot be directly sampled, and an alternative procedure is necessary. One option is importance sampling (IS), a method in which a distribution g on E (referred to as the importance distribution) is sampled from instead of π and the points $x_1, \ldots, x_N$ are assigned weights $w_i \propto \pi(x_i)/g(x_i)$ to compensate for the differences. These weighted points are referred to as particles. More formally, IS exploits the identities

$$\mathbb{E}_\pi(\psi) = Z^{-1} \int_E \psi(x)w(x)g(x)dx,$$

$$Z = \int_E w(x)g(x)dx,$$

to approximate $\mathbb{E}_\pi(\psi)$ using

$$\mathbb{E}_\pi(\psi) \approx \frac{\sum_{i=1}^{N} w_i \psi(x_i)}{\sum_{i=1}^{k} w_i}. \tag{2.1}$$

IS is also applicable in instances where only $\gamma(x) = \pi(x) \cdot Z$ can be evaluated as a factor of Z would be present in both numerator and denominator of Equation 2.1. These scenarios arise often in Bayesian inference.

The success of IS depends heavily upon the choice of g; it is beneficial for g to be as close to π as possible, as the variance of an estimation is approximately proportional to $1 + \mathrm{var}(w_i(x_i))$ [21]. Finding an appropriate g which can be tractably sampled from is often difficult, especially when X is non-standard and high-dimensional.

A possible solution is sequential importance sampling (SIS), an IS strategy in which the proposal is developed via recursive sampling of each component of

X conditionally on earlier components. That is, the variable $X = (X_1, \ldots, X_k)$ is sequentially sampled by individually proposing each X_i conditionally on $X_{1:(i-1)}$ using some sequence of kernels q_i. The result is an importance distribution g composed of the proposal distributions $g(X_{1:k}) = q_1(x_1) \prod_{i=2}^{k} q_i(x_i | x_{1:(i-1)})$.

Another possible solution is to use to use a variant of SIS called Sequential Monte Carlo (SMC) which simultaneously performs SIS on all N particles as a single batch. This facilitates particle resampling at intermediate steps, allowing the algorithm to focus on particles that have high weights (i.e. are representative of the target distribution), and discard those with low weights (i.e. are not representative of the target distribution). Resampling requires calculation of particle weights at these intermediate steps, thereby necessitating intermediate distributions $\pi_1, \ldots, \pi_k = \pi$ defined with the support of π_i being $x_{1:i}$.

The simplest resampling procedure involves multinomial sampling using the particle weights. However, other resampling mechanisms are available [11]. Often, resampling is not performed at every step. Instead the effective sample size (ESS [21]) is monitored and resampling is performed only when it drops below a pre-specified threshold. This generalization is sometimes given a separate name (such as SMC with adaptive resampling [11]). Here, no such distinction is necessary.

To ensure acceptable performance during the resampling step, it is desirable to use intermediate distributions π_i's which are similar to π, but on a restricted domain. In many popular applications of SMC (e.g. online problems such as particle filtering or time series), good intermediate distributions $\pi_1, \ldots, \pi_k$ are naturally provided by the layout of the problem (the posterior distribution up to time i). Otherwise, the user must design the intermediate distributions themselves. Like IS, the applicability of SMC extends to cases in which only $\gamma(x) \propto \pi(x)$ and $\gamma_i(x) \propto \pi_i(x)$ can be evaluated point-wise.

Algorithm 1 provides an algorithmic formulation of SMC employing N particles to approximate a k-dimensional distribution π. Here, $q_n(\cdot | x_{1:n-1})$ denotes the sequence of proposal distributions ($n \in 1:k$), the unnormalized intermediate densities are denoted $\gamma_{1:k}$, and $\gamma_k = \gamma$ represents the unnormalized target density. The variables $W_{1:n}^{(1:N)}$ are the normalized weights of the particles at each step, with $\tilde{w}_n^{(i)}$ denoting the weight update for particle i at step n.

We are now prepared to advance to Section 2.2, where Sequential Monte Carlo

Algorithm 1 : **SMC**

for $i = 1, \ldots, N$ **do**
  $X_1^{(i)} \sim q_1(\cdot)$
  $w_1^{(i)} \leftarrow \gamma_1(X_1^{(i)}) / q_1(X_1^{(i)})$
  $W_1^{(i)} \leftarrow w_1^{(i)} / \sum_{j=1}^{N} w_1^{(j)}$
end for
for $n = 2, \ldots, k$ **do**
  if the effective sampling size, $(\sum_i^N (W_{n-1}^{(i)})^2)^{-1}$, is smaller than a threshold T: **then**
    Resample the particles
    $W_{n-1}^{(i)} \leftarrow 1/N$ for all i
  end if
  for $i = 1, \ldots, N$ **do**
    $X_n^{(i)} \sim q_n\left(\cdot | X_{1:(n-1)}^{(i)}\right)$
    $\tilde{w}_n^{(i)} \leftarrow \gamma_n(X_{1:n}^{(i)}) / \left(\gamma_{n-1}(X_{1:n-1}^{(i)}) q_n(X_n^{(i)} | X_{1:(n-1)}^{(i)})\right)$
    $W_n^{(i)} \leftarrow W_{n-1}^{(i)} \tilde{w}_n^{(i)} / \sum_{j=1}^{N} W_{n-1}^{(j)} \tilde{w}_n^{(j)}$
  end for
end for

samplers, a non-standard special case of SMC algorithms, are discussed.

2.2 Sequential Monte Carlo Samplers

In classical Sequential Monte Carlo (SMC) setups, the dimensionality of the particles increases at each SMC iteration. For example, in a state-space or HMM model where each latent state is a univariate random variable, the particles are sequences of latent states that grow by one dimension at each step. A new particles at iteration n is constructed from an old one from iteration $n-1$ by copying the exact same first $n-1$ latent variables, and proposing one additional latent variable.

In contrast to classical SMC, SMC *samplers* [9] (SMCS) allow the dimensionality of the particles to be constant across SMC iterations. This facilitates the application of SMC to instances in which the proposal is intended to move complete particles through a sequence of distributions, such as in annealed importance sampling [23]. This becomes critical when one wants to use proposals inspired

from the Markov Chain Monte Carlo literature, where latent variables are modified rather than appended. Because of this flexibility, implementation of SMCS differs in an important way compared to classical SMC; the weight calculation is altered by an auxiliary backward kernel.

The general goal of SMCS is to sample from a probability measure $\mathrm{d}x$ defined on a probability space $(E, \mathscr{E})$. To do so, auxiliary distributions π_n, $n \in \{1, \ldots, p\}$ are introduced, where $\pi_p = \pi$. Abusing notation, assume that π_n has a density $\pi_n(x_n) = \gamma_n(x_n)/Z_n$, $x_n \in E$, with respect to a σ-finite dominating measure denoted $\mathrm{d}x$. Note that each x_n might itself be a vector, $x_n = (x_n(1), \ldots, x_n(k))$.

At a high-level, SMCS follow the structure of standard SMC algorithms very closely, with the important difference of a different weight update. As in the standard SMC case: We initialize the first iteration of the algorithm using the output of an importance sampling algorithm, $x_1^{(1:N)} = (x_1^{(1)}, \ldots, x_1^{(N)})$ with corresponding unnormalized weights $W_1^{(1:N)}$, where N denotes the number of particles. After initialization, we go from one iteration to the next using a proposal K_n, followed by a re-weighting, followed by an optional resampling step.

In the remainder of this section, we focus on the motivation behind the alternate weight update in SMCS. The intuition is that since the space does not grow at each iteration, it is no longer true that a particle at a given iteration has a unique possible ancestor in the previous generation. Unique ancestry provides a significant computational convenience for SMC algorithms. Otherwise, the calculation of importance weights involves a marginalization (integration) over all possible ancestor particles. Unique ancestry is true in classical SMC because particles are constructed by adding a suffix. It no longer holds once MCMC-type moves are used to perturb particles, as the several sequences of perturbations can result in the same destination.

SMCS circumvents the issue of non-unique ancestry by introducing auxiliary variables. This allows SMCS to be interpreted and analyzed as a standard SMC algorithm on a transformed space. More precisely, SMCS employs a new sequence of extended intermediate distributions $\{\tilde{\pi}_n\}$ defined such that $\tilde{\pi}_n$ has support E^n but admits π_n as a marginal. These intermediate distributions can be conceptualized as distributions over particle trajectories. Each perturbation increases the length of a particle trajectory. As a result, a problem with constant dimension has been

transformed into one of increasing dimension, a situation in which traditional SMC is applicable. The integral part of SMCS is that the sequence $\{\tilde{\pi}_n\}$ is defined such that π_n is admitted as a marginal. This ensures that the final weights of the particle trajectories approximate π_n without any additional computation or integration.

These intermediate distributions are designed using artificial backward Markov kernels $L_{n-1} : E \times \mathscr{E} \to [0,1]$ with densities $L_{n-1}(x_n, x_{n-1})$. Specifically,

$$\tilde{\pi}_n(x_{1:n}) = \tilde{\gamma}_n(x_{1:n})/Z_n$$
$$\tilde{\gamma}_n(x_{1:n}) = \gamma_n(x_n)\prod_{j=1}^{n-1} L_j(x_{j+1}, x_j).$$

Notice that $\tilde{\pi}_n$ admits π_n as a marginal. Each trajectory $x_{1:(n-1)}^{(i)}$ is propagated to $x_{1:n}^{(i)}$ using a proposal kernel $K_{n-1}(x_{n-1}, x_n)$. This results in a recursive expression for particle weights given by

$$w_n(x_{1:n}) = w_{n-1}(x_{1:(n-1)}) \cdot \tilde{w}_n(x_{n-1}, x_n) \tag{2.2}$$
$$\tilde{w}_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n) L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)}. \tag{2.3}$$

Given this formulation, we can now express the SMCS framework algorithmically, as given by [9], as Algorithm 2. Here, η_1 denotes the initial distribution from which the particles are drawn, with $\eta_1(x)$ denoting its density.

The output of Algorithm 2 is a collection of particles $X_{1:p}^{(1:N)}, W_p^{(1:N)}$ with each particle consisting of a trajectory and a weight. Because $\tilde{\pi}_p$ admits π_p as a marginal, π_p can be unbiasedly approximated using the marginalized particles $X_p^{(1:N)}, W_p^{(1:N)}$. This concludes the SMCS review.

It is important to recognize the power of the generality of the formulation of SMCS. Given this flexible framework, Chapter 3 is primarily a description and motivation of a particular formulation of proposals, intermediate distributions, and backward kernels which result in a particular class of SMCS's (referred to as DrSMC) which can handle problems involving deterministic relationships. Before moving into this detailed exposition, it is useful to review Hamiltonian Monte Carlo, as it plays a key role in the development of efficient proposals for DrSMC.

Algorithm 2 : **SMCS**

for $i = 1, \ldots, N$ **do**
  $X_1^{(i)} \sim \eta_1$
  $w_1(X_1^{(i)}) \leftarrow \gamma_1(X_1^{(i)}) / \eta_1(X_1^{(i)})$
end for
for $n = 2, \ldots, p$ **do**
  if the effective sampling size, $(\sum_i^N (W_{n-1}^{(i)})^2)^{-1}$, is smaller than a threshold T: **then**
    Resample the particles
    $W_{n-1}^{(i)} \leftarrow 1/N$ for all i
  end if
  for $i = 1, \ldots, N$ **do**
    $X_n^{(i)} \sim K_n(X_{n-1}^{(i)}, \cdot)$
    $\tilde{w}_n^{(i)} \leftarrow \tilde{w}_n(X^{(i)})$ (Equation (2.3))
    $W_n^{(i)} \leftarrow W_{n-1}^{(i)} \tilde{w}_n^{(i)} / \sum_{j=1}^N W_{n-1}^{(j)} \tilde{w}_n^{(j)}$.
  end for
end for

2.3 Hamiltonian Monte Carlo

Consider a k-dimensional continuous random variable $X_{1:k}$ whose distribution π admits a differentiable density. Random walk Metropolis-Hastings (MH) is a traditionally popular approach for approximating such distributions. Unfortunately, the performance of random walk Metropolis-Hastings (MH) worsens as k grows. The larger the dimension, the higher the likelihood that the proposal is poor in at least one of the dimensions of π. To compensate, the variance of the random walk is typically decreased as k grows, resulting in the speed of the exploration being unfeasibly slow. The computational cost per independent sample in random walk MH is $O(k^2)$ [8]).

In many of these cases, this curse of dimensionality can be mitigated through the use of Hamiltonian Monte Carlo (HMC) [22]. Let $\mathscr{L}(X_{1:k})$ denote the logarithm of this density. Hamiltonian Monte Carlo (HMC) is a specialized auxiliary variable Markov Chain Monte Carlo (MCMC) which incorporates information about the gradient of $\mathscr{L}(X_{1:k})$ to construct a proposal which favours moves towards high density areas. The results is far-off proposals that maintain high acceptance

probability. The cost per independent sample in HMC is approximately $O(k^{5/4})$ [19], allowing for better scaling to high dimensions than random walk MH. In addition, HMC's use of gradient information results in proposals which can capture highly correlated variables and other complicated dependencies.

HMC is based on a concept from physics known as Hamiltonian dynamics, meaning it has a physical interpretation which provides intuition for the workings of the algorithm. Consider the following analogy for HMC sampling. Suppose a ball is rolling around in an irregularly shaped bowl. An observer stands over the bowl taking photos of the ball at regular intervals. The camera is designed such that when a photo is snapped, a poof of air is expelled from the camera, randomly altering the ball's momentum. HMC is an algorithm that leverages the fact that, under certain conditions of the system, the proportion of time the ball spends in a region of the bowl is exponentially proportional to that region's volume. Thus, given an appropriately shaped bowl, the location of the ball resultant series of photos could be used to approximate a distribution of interest. The "certain conditions" are that the movement of the ball in the bowl is governed by Hamiltonian dynamics, and that the poofs of air update the momentum according to a standard normal distribution.

The above system can be viewed as an MCMC algorithm over an augmented space. Consider a more general setting where bowl exists in a $(k+1)$-dimensional space (with the $k+1$th dimension being the depth of the bowl). At any given point, the state of the ball can be characterized by k real-valued location variables, and k real-valued momentum variables. Let $X^t_{1:k}$ denote the location variables at time t, and $S^t_{1:k}$ denote the corresponding momentum variables. The depth of the bowl at a location $X_{1:k}$ is given by $\mathscr{L}(X_{1:k})$. According to Hamiltonian dynamics, the total "energy" of the ball at time t is given by

$$\mathscr{H}(X^t_{1:k}, S^t_{1:k}) = -\mathscr{L}(X^t_{1:k}) + \mathscr{K}(S^t_{1:k}), \tag{2.4}$$

where the summands represent "potential energy" (height of the ball) and "kinetic energy" (a function of momentum), respectively. Typically, the energy function is given by $\mathscr{K}(S^t_{1:k}) = \sum_{n=1}^{k} X_n^2/2$. The potential energy is the negative log density of π, and the kinetic energy is the negative log density of a standard k-dimensional

normal distribution. Thus, Equation 2.4 can be interpreted as the negative log density of a joint distribution on $X^t_{1:k}$ and $S^t_{1:k}$. It admits π as a marginal, and is the target density of the HMC algorithm.

The movement of the ball in between photos can be interpreted as a deterministic MCMC proposal. It is governed by a system of partial differential equations called Hamilton's equations. These are shown below as Equations 2.5 and 2.6.

$$\frac{\mathrm{d}S^t_n}{\mathrm{d}t} = -\frac{\partial \mathscr{H}}{\partial X^t_n} \tag{2.5}$$

$$\frac{\mathrm{d}X^t_n}{\mathrm{d}t} = \frac{\partial \mathscr{H}}{\partial S^t_n} \tag{2.6}$$

This movement conserves energy, meaning that $\mathscr{H}(X^t_{1:k}, S^t_{1:k})$ (and therefore the target density) is invariant to the ball's rolling. This is a desirable property for a MCMC proposal, as it eliminates the need of a MH rejection step. However, Hamiltonian motion alone is not enough to build a valid MCMC proposal, as the chain would be restricted to a single contour of the density. HMC achieves full ergodicity through the second momentum refreshing ("air poof") step. That is, at fixed time points (after each photo) a Gibbs' step is applied to the momentum variables, resampling them from their standard normal marginals.

In situations where the Hamiltonian mechanics can be integrated analytically, the result is an MCMC algorithm capable of far off proposals with no MH rejection step. Unfortunately, except for special cases such as in [24], Hamilton's equations cannot be integrated exactly. Instead, standard practice is to discretize the movement using a "leapfrog" integrator [22]. Given a discretization step-size ε, a leapfrog update is given by the following function,

Leapfrog$(X_{1:n}, S_{1:n}, \varepsilon)\{$

$\tilde{S}_{1:n} \leftarrow S_{1:n} + (\varepsilon/2)\nabla_{X_{1:n}} \mathscr{L}(X_{1:n})$

$\tilde{X}_{1:n} \leftarrow X_{1:n} + \varepsilon \tilde{S}_{1:n}$

$\tilde{S}_{1:n} \leftarrow \tilde{S}_{1:n} + (\varepsilon/2)\nabla_{\tilde{X}_{1:n}} \mathscr{L}(\tilde{X}_{1:n})$

return $(\tilde{X}_{1:n}, \tilde{S}_{1:n})$ $\}$ where ∇_θ denotes the gradient with respect to θ. The interval of "time" between samples is controlled by ε and the number of leapfrog steps. Note that "time" is an artificial computational construct which bears no direct relationship with physical time.

A standard HMC proposal consists of a Gibbs resampling the momentum followed by a simulation of the particle moving according to Hamiltonian dynamics for L leapfrog steps. The inexact nature of the leapfrog steps necessitates the introduction of a MH rejection step to maintain invariance of $\exp(H(\theta, p))$. Algorithm 3 summarizes a standard implementation of an HMC proposal. Here, $\mathcal{N}(0, I)$ denotes a multivariate normal distribution (with mean 0 and identity covariance matrix) and unif$(0, 1)$ denotes a uniform distribution on the interval $[0, 1]$.

Algorithm 3 : **HMC**

Inputs: $X_{1:n}$, ε, L, $\mathscr{H}$;
  $\tilde{X}_{1:n} \leftarrow X_{1:n}$
  Draw $S_{1:n} \sim \mathcal{N}(0, I)$
  $\tilde{S}_{1:n} \leftarrow S_{1:n}$
  for $i = 1, \ldots, L$ **do**
    $\tilde{X}_{1:n}, \tilde{S}_{1:n} \leftarrow$ Leapfrog($\tilde{X}_{1:n}$, $\tilde{S}_{1:n}$, ε)
  end for
  Draw $u \sim$unif$(0, 1)$
  $MH \leftarrow \exp(\mathscr{H}(X_{1:n}, S_{1:n}) - \mathscr{H}(\tilde{X}_{1:n}, \tilde{S}_{1:n}))$
  if $u < MH$ **then**
    return $(\tilde{X}_{1:n}, \tilde{S}_{1:n})$
  else
    return $(X_{1:n}, S_{1:n})$
  end if

Reversibility of the HMC proposal holds due to the negation the momentum at the end of the simulation. In practice, this negation need not be performed because it is immediately followed by a Gibbs resampling of $S_{1:n}$. HMC can still be valid if the support of θ is constrained, as long as the support is connected (e.g. $\theta_d \geq 0$). If a leapfrog step violates the constraint, it is standard to negate p and reflect θ across the violated constraint boundary.

The performance of HMC depends heavily on choice of ε (the step size) and L (the number of steps). If ε is chosen too large, the associated discretization error results in an excessive proposals rejection rate. Moreover, a choice of ε that is too small leads to wasted computation time. A too small value of L will result in successive proposed values being too close to each other. If L is too large, the particle's trajectory form a loop, also resulting in wasted computation.

Heuristics for choosing ε and L, as well as strategies for tuning them by hand, are discussed in [22]. Unfortunately, this tuning is time consuming, often involving multiple preliminary runs and may require expert knowledge. Recently, there has been progress in automatically tuning HMC proposals [19, 32]. As will become apparent in Chapter 3, DrSMC represents a particularly difficult case for tuning ε. Fortunately, I have developed a specialized leapfrog integrator based on split Hamiltonian Monte Carlo [26] for DrSMC on problems involved deterministic sums. I elaborate on this approach in Chapter 3.

Chapter 3

DrSMC

DrSMC is a new Sequential Monte Carlo Sampler (SMCS) for continuous random vectors conditioned on a deterministic constraint. This chapter is a presentation of the DrSMC algorithm which is organized as follows. Section 3.1 outlines a broad class of deterministically constrained Bayesian inference problems to which DrSMC can be applied. Sections 3.2 – 3.4 define DrSMC in terms of its SMCS components. Section 3.1 formulates the intermediate distributions, Section 3.2 provides the form of backward kernels, and Section 3.4 discusses several choices for the proposal kernels. In this Section, emphasis is placed on the exposition of a specialized proposal kernel developed specifically for problems involving deterministic sums. A general algorithmic statement of DrSMC is provided in Section 3.5. Section 3.6 discusses how to exactly (rather than approximately) enforce deterministic constraints using DrSMC for deterministic sum problems.

3.1 Problem Formulation and General Idea

Consider the problem of conducting inference on a model containing latent variables $X_{1:k}$ with a joint density $p(x)$. We are interested in computing expectations of some test functions h (for example, indicator variables, moments, etc), that condition on satisfaction of a deterministic constraint: $\mathbb{E}[h(X)|Y,G]$, where $G = f(X) - s$, and $G = 0$.[1] Here, $f : \mathbb{R}^k \to \mathbb{R}$ is a function encoding a constraint, assumed to be

[1] Formally defining this conditional expectation requires some care to avoid conditioning on an event of zero probability. To be more precise, what we wish to compute is a version of a density

continuous, and $s \in \mathbb{R}$ is assumed to be known. We call G the random Lagrangian, in parallel to the form and function of the classical Lagrangian in the method of Lagrange multipliers from optimization.

DrSMC approximates the posterior distribution $X|G$ by generating a collection of N particles $\left\{(x^{(1)}, w^{(1)}), \ldots, (x^{(N)}, w^{(N)})\right\}$ which satisfy the deterministic relation, either exactly or approximately, depending on the precise form of f—this is discussed in Section 3.6. The intuition behind DrSMC is that the particles in the first SMC iteration approximate the distribution $p(x)$ (drawn exactly from $p(x)$ if it is easy to sample, otherwise obtained through importance sampling), and then these particles are propagated through a series of intermediate distributions to gradually introduce the constraint $G = 0$.

Note that it is possible to use SMCS (and therefore DrSMC) to gradually introduce other attributes of $p(x)$, such as a likelihood term [7], multi-modality [23], or monotone constraints [17]. In theory, these approaches can be combined with DrSMC to simultaneously introduce several aspects the target distribution. However, tuning such algorithms is a very difficult problem in practice. Empirically, we have achieved promising results in some small examples through time-consuming pilot runs. However, developing an automatic tuning strategy which generalizes to many instances is still an open problem. For this reason, here I focus on an exposition of DrSMC which solely introduces a constraint.

3.2 Intermediate Densities

Let $\varphi_{\sigma^2}(x)$ denote a normal density with mean 0 and variance σ^2. DrSMC gradually enforces the deterministic relationship by defining intermediate distributions π_n with the following densities:

$$\gamma_n(x) = p(x)\varphi_{b_n}(f(x) - s)$$

where $b_{1:p}$ is a positive decreasing sequence. This annealing approach ensures that a variety of particles following $p(x)$ are considered, but as the algorithm progresses through the intermediate distributions, the weight updates and resampling

of $\mathbb{E}[h(X)|G]$, evaluated at $G = 0$, and selected in the same fashion as in the standard argument for justifying conditioning on a continuous observation in Bayesian models.

steps cause DrSMC to hone in on particles which approximately satisfy the constraint and have high probability. DrSMC's performance is sensitive to choice of $b_{1:p}$. Annealing too quickly causes the ESS to plummet which results in particle degeneracy. Annealing too slowly results in wasted computation. In Section A.2, it is shown that under reasonable assumptions, a good functional form for $b_{1:p}$ is

$$b_n = \alpha\beta^{-n},$$

where α and β are both real numbers such that $\alpha > 0$ and $\beta > 1$. The use of the normal density is valid even if the range of f is only a subset of the real numbers; it amounts to an truncated normal density.

Notice that, for finite p, γ_p is not the exact posterior distribution. Particles following these densities do not exactly satisfy the deterministic relationship. However, they can be made arbitrarily close to the exact target distribution by making b_p large enough. There are situations in which it is possible to enforce the relationship exactly with a final rounding step, while preserving the correct target distribution. We detail this procedure in Section 3.6.

3.2.1 Comparison to SCMC

The use of SMCS intermediate distributions to enforce a deterministic constraint was independently developed in [17] as a specialized application of Sequentially Constrained Monte Carlo (SCMC). However, SCMC uses a different formulation of intermediate distributions. There, the intermediate distributions are given by

$$\gamma'_n(x) = p(x)\Phi(-\tau_n|f(x) - s|),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution, and $\tau_{1:p}$ is an increasing sequence of non-negative numbers. However, this series of intermediate distributions lack some of the advantages of those employed in DrSMC. The functional form of the optimal sequence of $\tau_{1:p}$ is unknown for problems involving deterministic constraints, making it much more difficult and time-consuming to determine a sequence $\tau_{1:p}$ which controls the descent of the effective sample size. Additionally, HMC-based proposal kernels discussed in

Section 3.4 would require additional developments, because γ_n' is not differentiable everywhere.

3.3 Backward Kernels

Typically, a good measure of the performance of a SMCS is the variance of the unnormalized importance weights w_n, with a lower variance indicating better performance. Using this measure, given the proposal kernels, the optimal backward kernel [9] of an SMCS is of the form

$$L_{n-1}^{\text{opt}}(x_n, x_{n-1}) = \frac{\eta_{n-1}(x_{n-1})K_n(x_{n-1}, x_n)}{\eta_n(x_n)}$$

where η_n is the marginal importance distribution at time n and $K_{1:p}$ is the sequence of proposal kernels. Unfortunately, calculating $\eta_n(x_n)$ point-wise is often intractable as it involves marginalizing over all possible paths that end in x_n. It is therefore customary to use suboptimal backward kernels in practice [9].

DrSMC employs a backward kernel of the form

$$L_{n-1}(x_n, x_{n-1}) = \frac{\gamma_n(x_{n-1})K_n(x_{n-1}, x_n)}{\gamma_n(x_n)}.$$

Assuming that $\gamma_n(x_{n-1})/\gamma_n(x_n)$ is well approximated by $\eta_{n-1}(x_{n-1})/\eta_n(x_n)$, such a backward kernel is a tractable approximation of the optimal kernel. This backward kernel first described by [23] as a justification of the validity of Annealed Importance Sampling. It was also given special attention by [9] because it reduces the weight update calculation in Equation 2.3 to

$$\tilde{w}(x_{n-1}, x_n) = \gamma_n(x_{n-1})/\gamma_{n-1}(x_{n-1}).$$

Given the intermediate distributions $\gamma_{1:p}$ provided in Section 3.2, it can be further reduced to

$$\tilde{w}_n(x_{n-1}, x_n) = \exp\left(\left(b_{n-1}^{-1} - b_n^{-1}\right)\left(f(x_{n-1} - s)^2\right)\right). \tag{3.1}$$

Such an expression indicates that the weight of a particle at step n is indepen-

dent of its current value x_n given x_{n-1}. As is discussed in Section 3.5, there are computational benefits of such an independence. Note that DrSMC's backward kernel formulation is only valid when $K_n(x_{n-1}, x_n)$ leaves π_n invariant. Otherwise, DrSMC may not admit the target distribution as a marginal. This necessitates that DrSMC employ proposal kernels which satisfy this property. Section 3.4 elaborates on developing such kernels.

3.4 Proposal Kernels

Building suitable π_n invariant proposal kernels K_n is the most difficult part of developing a DrSMC. Here, we consider two directions used to formulate kernels: Metropolis-Hastings and Hamiltonian Monte Carlo. The success of these proposals rely heavily on tuning parameters. For many instances, developing efficient tuning approaches remains an open problem. Instead of focusing on the negative, the emphasis in this section is placed on a specialized kernel formulation for problems involving deterministic sums.

3.4.1 Metropolis-Hastings Proposals

Given their popularity for Markov chain Monte Carlo (MCMC), it is reasonable to first consider Metropolis-Hastings (MH) when developing π_n invariant kernels. Indeed, such proposals are the standard in many SMCS (such as SCMC). Moving a particle according to a MH kernel consists of two steps. First, a new value x_{n+1} for particle is proposed (conditional on the current value x_n) according to the density $\rho_{n+1}(x_{n+1}|x_n)$. Then, the proposal is accepted if and only if

$$u < \frac{\gamma_{n+1}(x_{n+1})\rho_{n+1}(x_n|x_{n+1})}{\gamma_{n+1}(x_n)\rho_{n+1}(x_{n+1}|x_n)},$$

where u is randomly generated from a uniform(0,1) distribution. This accept/reject step ensures that K_n leaves π_n invariant. Point-wise evaluation of such K_n kernels can be computationally intensive, mainly due to the rejection step. Evaluating $K_n(x_n, x_n)$ involves marginalizing over all possible rejected proposals given a starting point. Our choice of backward kernel results in a SCMC weight update which does not involve point-wise evaluation of K_n thereby avoiding these potential dif-

ficulties. However, there are other obstacles which can result in poor performance of DrSMC under a MH proposal kernel.

Typically, $\rho_{n+1}(x_{n+1}|x_n)$ consists of a step in a random walk[2]. Due to their simplicity, random walk MH kernels yield algorithms for which the computational cost per particle is low. However, many particles and steps are usually needed in practice since MH moves are relatively uninformed.

As DrSMC progresses (n increases), the density of γ_n is increasingly concentrated on a small region where the constraint is approximately satisfied. This means that the acceptance rate of uninformed far-off proposals is decreasing, thus necessitating smaller and smaller suggested moves as n progresses. This requires tuning of the sequence of ρ_n. In addition, smaller and smaller proposed moves result in the proposal's exploration of the space being very inefficient once n gets large, regardless of how well it is tuned. As the dimension k of a problem increases, these problems become more and more pronounced. Correlated variables in X further exacerbate the issue. This motivates another direction of proposals which are more informed.

3.4.2 HMC Proposals

Scaling up to high-dimensional problems can be made much easier through the use of informed moves. If the target density is differentiable, informed moves can be developed by leveraging gradient information, such as in HMC. While the usage of HMC in an MCMC framework is increasingly popular, so far the use of HMC in a SMC framework has been limited. There has been previous work [6, 25, 28] in which HMC has been employed in a sequential sampling framework, but none within an SMCS framework. In [6, 25], HMC is implemented within SMC (not SMCS; i.e. an expanding state), and [28] describe HMC within Annealed Importance Sampling which contains no resampling step. As far as I know, HMC within SMCS is a novel contribution of this thesis.

My formulation of HMC as a SMCS proposal kernel follows a similar format to an implementation of HMC in an MCMC context (as described in Algorithm 3). First, the auxiliary momentum variables s_n are instantiated by drawing from a mo-

[2]Here, our discussion concerns such proposals. Other choices from the MCMC literature can be used, and may negate such issues, depending on the specifics.

mentum distribution. Then, the Hamiltonian movement is simulated (typically using the leapfrog integrator) to obtain the proposed (x'_n, s'_n). Finally, a MH rejection step is applied. The variable x_{n+1} is obtained by discarding the momentum variables after the MH step. Note that the new instantiation and then discard of the momentum variables within each step is what separates HMC within SMCS from HMC within MCMC.

The above process amounts to a non-traditional MH kernel. Therefore, many of the benefits of a MH kernel are retained, such as invariance of π_n and the avoidance of evaluating K_n. In addition, these kernels enjoy the benefits of HMC, such as far-off proposals with high acceptance probability. In terms of the bowl analogy introduced in Section 2.3, the sequence of distributions described by $\gamma_{1:p}$ correspond to bowls which become increasingly deep at values of x where the random Lagrangian $G = 0$. Therefore, Hamiltonian proposals have the benefit of systematically preferring moves which advance the particle toward satisfying the constraint. As a result, DrSMC with HMC proposal kernels require far fewer steps to satisfy the constraint in practice than those with MH proposal kernels. In addition, the gradient of the log density of the random Lagrangian is often trivial to compute.

Unfortunately, Hamiltonian motion is usually very computationally intensive to simulate relative to random walks. This means that far fewer particles can be used than in DrSMC with MH proposal kernels. In addition, tuning HMC within SMCS is very difficult. The existing tuning methods in[19, 32] are intended for tuning HMC as part of a MCMC strategy, not within a SMCS strategy such as in DrSMC. Tuning a HMC proposal for SMCS is a fundamentally different problem; MCMC strategies involve simulating a single Markov Chain from a single target distribution, whereas SMC samplers involve simulating multiple particles across a series of different intermediate distributions. The existing SMC with HMC kernel literature [6, 28] contains no discussion of automatic tuning procedures.

In practice, the appropriate value for ε will differ across intermediate distributions. If a naive leapfrog integrator is used, the latter steps of DrSMC require a value of ε so small that decent proposals are effectively impossible. This is due to the fact that as n increases, the Hessian of the random Lagrangian becomes more and more extreme. This necessitates smaller and smaller values of ε as n increases in order to keep the approximation accurate enough to have accepted MH moves.

This "vanishing epsilon" problem means that HMC kernels based solely on the leapfrog integrator are impractical at the latter steps of the DrSMC algorithm.

The desire to eliminate the need of vanishing epsilons motivated my development of the split HMC proposal kernel for DrSMC, which I now describe in Section 3.4.3. Note that thus far, my developments are only applicable to problems for which the deterministic relationship can be expressed as a sum.

3.4.3 Split-HMC Proposal for Deterministic Sums

To facilitate the split-HMC formulation which eliminates the vanishing epsilon problem, I will briefly summarize the general mechanics of split-HMC, and then discuss the ingredients of the specialized split-HMC formulation.

Although exact simulation of Hamiltonian motion is typically intractable, there are special cases (such as in [24]) for which it is possible to analytically solve Equations 2.5 and 2.6, allowing one to simulate the motion exactly. When such a solution is available, simulation is very efficient and there is no need to include an MH rejection step. Unfortunately, these special cases rarely arise in applications. However, a recent advance in the literature, called split Hamiltonian Monte Carlo, has made it possible to exploit the computational efficiency associated with exact simulation even when the target density is not one of the special cases.

In the split HMC [26] strategy for simulating Hamiltonian motion, the Hamiltonian energy function is "split" into two components, one of which admits an analytic solution. That is, $\mathscr{H}(X^t_{1:k}, S^t_{1:k})$ in Equation 2.4 is decomposed into two summands,

$$\mathscr{H}(X^t_{1:k}, S^t_{1:k}) = \mathscr{H}_1(X^t_{1:k}, S^t_{1:k}) + \mathscr{H}_2(X^t_{1:k}, S^t_{1:k}), \tag{3.2}$$

where $\mathscr{H}_2(X^t_{1:k}, S^t_{1:k})$ is a special case for which Hamiltonian motion can be simulated exactly. Typically, the decomposition is such that $\mathscr{H}_1(X^t_{1:k}, S^t_{1:k}) = -\mathscr{L}_1(X^t_{1:k})$ and $\mathscr{H}_2(X^t_{1:k}, S^t_{1:k}) = -\mathscr{L}_2(X^t_{1:k}) + \mathscr{K}(S^t_{1:k})$, with $-\mathscr{L}_1(X^t_{1:k}) - \mathscr{L}_2(X^t_{1:k})$ being the negative log likelihood of the target distribution. It is shown in [26] that in such cases, it is valid to integrate the Hamiltonian motion by replacing each leapfrog step with a hybrid of exact simulation of $\mathscr{H}_2(X^t_{1:k}, S^t_{1:k})$ and a leapfrog step on $\mathscr{H}_1(X^t_{1:k}, S^t_{1:k})$. This is expressed in the SplitLeapfrog$(X_{1:n}, S_{1:n}, \varepsilon)$ function shown below. Note that here, $\mathscr{E}^{\mathscr{H}_2}_{\varepsilon}(\cdot)$ is a function that returns the result of perfectly sim-

ulating Hamiltonian motion according to $\mathscr{H}_2$ for an ε time interval.

SplitLeapfrog$(X_{1:n}, S_{1:n}, \varepsilon)\{$
$\tilde{S}_{1:n} \leftarrow S_{1:n} + (\varepsilon/2)\nabla_{X_{1:n}}\mathscr{L}_1(X_{1:n})$
$(\tilde{X}_{1:n}, \tilde{S}_{1:n}) \leftarrow \mathscr{E}_\varepsilon^{\mathscr{H}_2}(\tilde{X}_{1:n}, \tilde{S}_{1:n})$
$\tilde{S}_{1:n} \leftarrow \tilde{S}_{1:n} + (\varepsilon/2)\nabla_{\tilde{X}_{1:n}}\mathscr{L}_1(\tilde{X}_{1:n})$
return $(\tilde{X}_{1:n}, \tilde{S}_{1:n})$ $\}$.

Because of the approximate component of the SplitLeapfrog integrator, it is still necessary to implement an MH rejection step after the proposed move. In [26], it is claimed that this approach performs well if $\mathscr{H}_2$ is a reasonable approximation of $\mathscr{H}$. Often, much larger values for ε can be employed in split-HMC than for the naive leapfrog integrator. This was my motivation for investigating split-HMC to eliminate the vanishing epsilon problem in DrSMC.

In the appendix (Section 6.1), I demonstrate that Hamiltonian motion can be simulated exactly for random Lagrangian terms which are expressed as $\sum_{i=1}^{k} X_i - s$. Therefore, for DrSMC problems involving deterministic sum constraints, it is possible to employ a split-HMC kernels as outlined above, with $\mathscr{L}_1$ being the log density of the prior and $\mathscr{L}_2$ being $-G^2/b_n$. This results in a SplitLeapfrog function which alternates between updating the momentum based on the prior and simulating motion according to the constraint.

Using this integrator instead of the naive leapfrog within a HMC-based K_n has several practical benefits. The expression of $\mathscr{H}_1$ does not involve b_n. As a result, ε need not depend on b_n, as ε need only be tuned to accommodate $\mathscr{H}_1$. Since $b_{1:p}$ is the only varying parameter in the sequence $\gamma_{1:p}$, tuning ε for DrSMC is no longer a moving target; $\mathscr{H}_1$ is the same from step to step. This allows a single ε to be used for the DrSMC full algorithm run, eliminating the vanishing epsilon problem. The proposal kernels continue to make far off proposals in latter stages of the algorithm without any increased computational cost or low acceptance probabilities. In addition, ε can now be tuned using the existing methods in the HMC for MCMC literature, e.g. [32].

This negates the major problems associated with HMC proposal kernels previously discussed. For this reason, I advocate the use of such split-Hamiltonian proposal kernels for instances in which random walk kernels perform poorly (e.g. high-dimensional problems with correlated variables).

3.5 DrSMC Algorithms

With the SMCS components (intermediate distributions, backward kernels, and proposal kernels) which characterize DrSMC defined, I can now state DrSMC as Algorithm 4. This statement is for a generic π_n invariant proposal kernel K_n, any of the three (or others) discussed in Section 3.4 are valid.

The choice of backward kernel described in Section 3.3 resulted in a weight update given by Equation 3.1. Such a weight update indicates that $\tilde{w}_n(X)$ is independent of X_n given X_{n-1}. As recommended by [9], this independence allows for a modification of the SMCS algorithm such that resampling occurs before propagation of the particles. This improves efficiency of the algorithm, as time is not wasted propagating particles that do not survive the resampling step. Algorithm 4 includes this modification.

Algorithm 4 : **DrSMC**

for $i = 1, \ldots, N$ **do**
  $X_1^{(i)} \sim \gamma_1$
  $w_1(X_1^{(i)}) \leftarrow 1/N$ for all i.
end for
for $n = 2, \ldots, p$ **do**
  for $i = 1, \ldots, N$ **do**
    $X_n^{(i)} \sim K_{n-1}(X_{n-1}^{(i)}, \cdot)$
    $\tilde{w}_n^{(i)} \leftarrow \tilde{w}_n(X_{n-1}^{(i)})$ (Equation (3.1))
    $W_n^{(i)} \leftarrow W_{n-1}^{(i)} \tilde{w}_n^{(i)} / \sum_{j=1}^{N} W_{n-1}^{(j)} \tilde{w}_n^{(j)}$.
  end for
  if If the effective sampling size, $(\sum_i^N (W_n^{(i)})^2)^{-1}$, is smaller than a threshold T:
  then
    Resample the particles
    $W_n^{(i)} \leftarrow 1/N$ for all i
  end if
end for

Note that the above DrSMC results in particles that approximately satisfy the constraint $G = 0$. For deterministic sum constraints, it is possible to ensure that the constraint is exactly satisfied. I elaborate on this point in Section 3.6.

3.6 Exact Enforcement

In DrSMC, as particles are propagated through the intermediate distributions $\gamma_{1:p}$, the values of the random Lagrangians associated with the particles converge to 0. However, unless additional steps are taken the constraint is not perfectly satisfied in finite time (i.e. $f(X) \sim N(s, b_p)$). When $f(X)$ encodes a sum constraint, augmenting the DrSMC to satisfy the constraint exactly is straightforward through the introduction of a $(p+1)$th SMCS step. However, this final step does not utilize the same form of intermediate distribution, backward kernel, or proposal kernel as the previous steps. Instead, they are defined as follows:

$$\gamma_{p+1}(x) = p(x)\mathbb{1}(f(x) = s),$$

where $\mathbb{1}(\cdot)$ is defined to be 1 if its argument is true, and 0 otherwise.

$$L_{p+1}(x_{p+1}, x_p) = \varphi_{b_p}(x_{p+1}(k) - x_p(k))$$

where $x_n(k)$ denotes the value of the kth variable in x_n.

$$K_{p+1}(x_p, x_{p+1}) = \delta(x_{p+1} = y(x_p)),$$

where $\delta(\cdot)$ is the dirac-delta distribution and

$$y(x_p) = \left(x_p(1), x_p(2), \ldots, y_p(k-1), s - \sum_{i=1}^{k-1} x_p(i) \right).$$

Note that this proposal kernel deterministically enforces the constraint. Although non-standard, this configuration of backward kernel and proposal distribution are valid, meeting the criteria outlined in [9]. They also result in several cancellations in the weight update, resulting in the final expression

$$\tilde{w}_{p+1}(x_p, x_{p+1} = p(x_{p+1})/p(x_p).$$

For DrSMC for deterministic sums, this final step can be implemented at the end of Algorithm 4. If desired, the ESS can be recalculated, followed by a resampling

step. Chapter 4 demonstrates DrSMC with a split-HMC proposal on a synthetic deterministic sum problem.

Chapter 4

Demonstration and Experiments

This chapter showcases the performance of DrSMC with split-HMC on a synthetic deterministic sum problem. The organization of the chapter is as follows. Section 4.1 formulates the problem. Section 4.2 both describes a DrSMC algorithm configuration for attacking the problem and illustrates experimental results obtained from simulation. Section 4.3 compares the performance of the DrSMC algorithm to that of two existing algorithms, Dynamic Scaled Sampling (DYSC [20]), and Sequentially Constrained Monte Carlo (SCMC [17])).

4.1 Problem Formulation

Consider a 15-dimensional random vector X with multivariate normal distribution $MVN(0,\Sigma)$. Here, $\Sigma = D\Omega D$ where Ω is a 15×15 correlation matrix defined such that

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } i = j \\ -0.6 & \text{if } i - j \text{ is odd} \\ 0.6 & \text{otherwise,} \end{cases}$$

and D is a 15×15 diagonal matrix with $D_{i,i} = \sqrt{(16-i)}$. Essentially, Ω encodes two distinct groups ($\{X_1, X_3, X_5, \ldots, X_{15}\}$ and $\{X_2, X_4, \ldots, X_{14}\}$) of variables with positive within-group correlation (0.6) and negative between-group correlations (-0.6). The marginal variances of $X_{1:15}$ decrease sequentially such that X_i possesses a marginal variance of $16 - i$.

The investigation in this chapter focuses on inferring the posterior distribution of the vector X described above given that $\sum_{i=1}^{15} X_i = 20$. This synthetic problem is an example of a deterministic sum problem for which DrSMC with split HMC proposals is well-suited. It is relatively high-dimensional, possesses correlation among variables, and involves inhomogeneity among the variables. An added bonus is that the true posterior distribution for this problem can be obtained analytically using special properties of the multivariate normal distribution (details regarding this derivation are provided in the appendix). The availability of the exact solution facilitates assessment of DrSMC's performance.

4.2 Results

For the problem outlined in Section 4.1, consider the DrSMC with split-HMC proposals algorithm configuration outlined in Table 4.1. Here, the number of steps and number of particles are balanced to achieve good performance in a computational runtime of five minutes. Note that the 31st step is an exact enforcement step as described in Section 3.6. The resampling threshold of 250 (half the number of particles) is standard practice in the SMC literature. The value of α is specified such that b_1 is approximately equal to the variance of the unconstrained $\sum_{i=1}^{15} X_i$, and the parameter β is assigned a value which was chosen using a short pilot run (it corresponds to $C = 0.8$ in Section 6.2). The HMC parameters are assigned values which were determined using pilot runs on the prior. They balance computational burden, high acceptance rate, and efficiency in exploring the space.

Diagnostic plots and indicators of performance for a run of the algorithm are provided as Figures 4.1– 4.4. Figure 4.1 provides a comparison of the approximated posterior with the true theoretical values. Section 6.3 provides the details on the derivation of the theoretical values. This plot visually demonstrates that the DrSMC algorithm accurately captures both the shape and mean of the marginal distributions of $X_{1:15}$.

Figure 4.2 shows the trajectory of effective sample size across steps, as well as the number of particles which survive each resampling step. Note that the effective sample size is reset each time resampling is performed. As desired, the ESS descends gradually. This evidence supports the adoption of an exponential decay

Table 4.1: Configuration of the DrSMC with HMC proposals used to approximate the distribution described in Section 4.1.

SMCS Parameters	
Number of Particles	500
Resample Threshold	250
Number of Steps	31
Determinants of $b_{1:p}$	
α	14.5
β	1.2026
HMC parameters	
ε	0.3
L	3

schedule of $b_{1:p}$. The plot also demonstrates the number of HMC proposals that are accepted at each step. Using the split-HMC proposal, one would expect the acceptance rate to remain relatively constant. The volatility of the acceptance rate of the proposal at later steps is unexpected, and requires further investigation.

Figure 4.3 shows the trajectory of $f(X)$ of the particles as they are propagated through the 30 steps. At step one, $f(X) < 20$ for all particles. This is to be expected, as the prior is centred at $f(X) = 0$. Despite this poor initialization with respect to the constraint, a combination of split-HMC proposals and resampling ensures that all particles approximately satisfy $f(X) = 20$ by step 30. The gradual convergence shown in Figure 4.3 is the hallmark of a healthy DrSMC algorithm. It demonstrates that both the proposal and resampling steps are helping to push the particles toward the constraint. This is in contrast to situations in which the proposal is poor. In such instances, almost all movement toward the constraint occurs due to resampling. In such cases, the $f(X)$ trajectory plot has a block-like structure with large declines at the resample steps. As a result, DrSMC performs poorly because the approximation is based on only a few initial ancestor particles.

Figure 4.4 provides a more complete view of how the particles satisfy the constraint at step 30. The density estimate of constraint before the exact enforcement step is as expected; it closely resembles a normal density centred at 20 with variance b_{30}. More measures of performance of DrSMC are available for multiple runs in Section 4.3.3, where it is contrasted with the performance of existing algorithms.

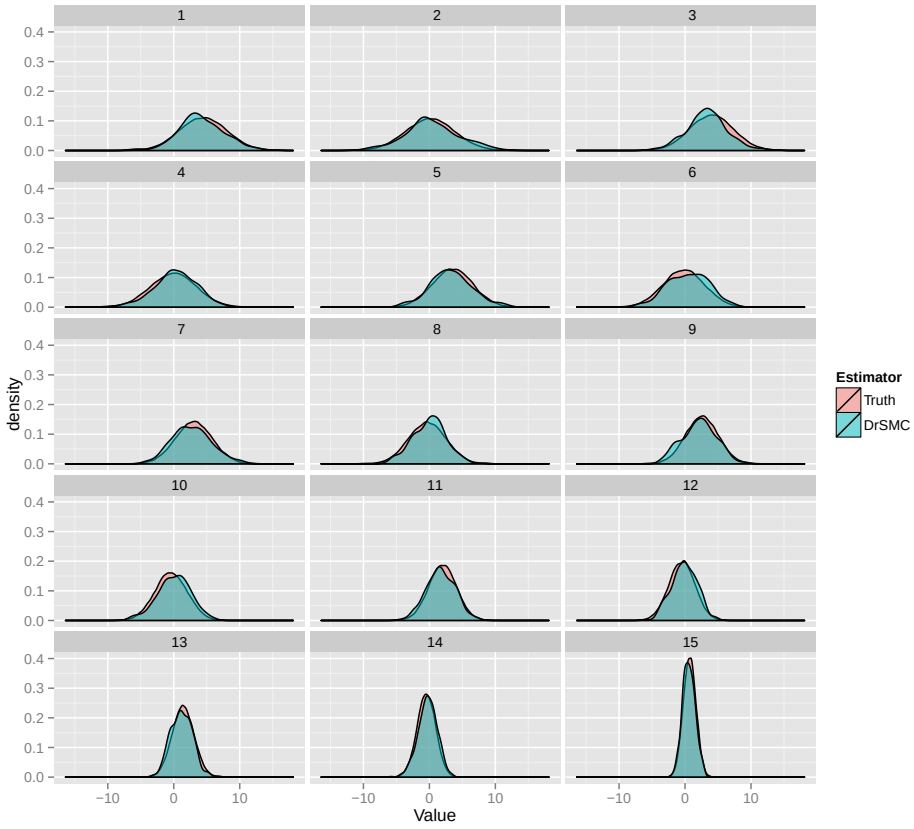

Figure 4.1: A comparison of DrSMC posterior density estimates (shown in red) and the true posterior densities (shown in green) for the 15-dimensional multivariate normal problem described in Section 4.1.

4.3 Numerical Comparisons to Previous Methods

In order to fully appreciate the strength of DrSMC with split-HMC proposals, it is useful to contrast performance with that of approaches in the literature. In this section, I compare the results of running DrSMC to that of two existing algorithms. The results of running each algorithm five times on the problem outlined in Section 4.1 are available in Section 4.3.3. Before presenting these results, I will briefly describe the configurations of the two existing algorithms in Sections 4.3.1 and 4.3.2.

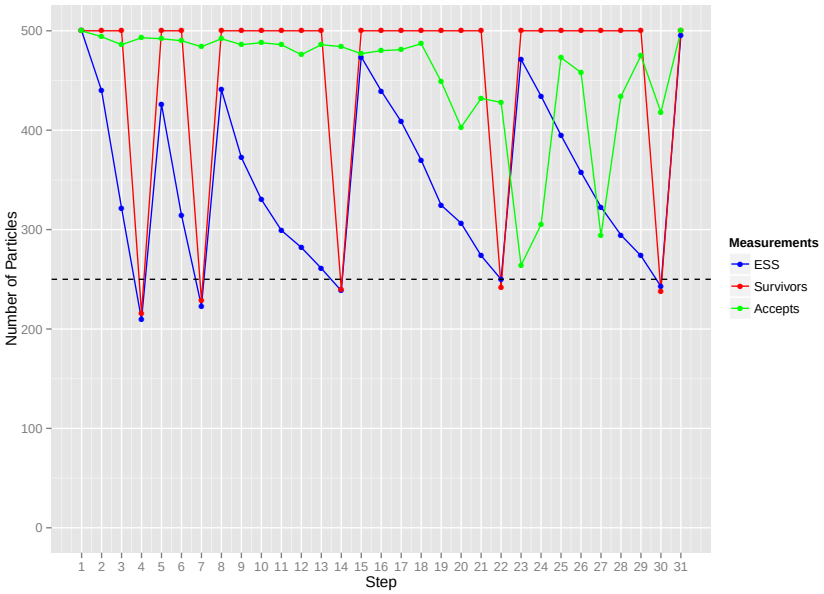

Figure 4.2: An illustration of the effective sample size trajectory (shown in blue) across 31 steps of a DrSMC approximation for the 15-dimensional multivariate normal problem described in Section 4.1. The red line depicts the number surviving particles after each resampling, and the green line reports the number of HMC proposals acceptance.

4.3.1 DYSC

Dynamic scaled sampling (DYSC) [20] is an importance sampling framework which targets the posterior distribution of a sequence of random variables $X_{1:k}$ conditioned $\sum_{i=1}^{k} X_i = s$. The DYSC procedure is characterized by sequentially transforming the proposal distributions to ensure that the expected sum satisfies the constraint. Although it is primarily intended for sums of independent (and identically distributed) non-negative random variables, the DYSC philosophy easily extends to more general settings, such as our synthetic problem described in Section 4.1.

The recipe for the general DYSC framework considered here is described as Algorithm 5, with $\gamma(\cdot)$ denoting a (potentially unnormalized) target distribution and $q_j(\cdot|\eta_j)$ denoting the marginal distribution of X_j translated such that its expectation is η_j. In [20], the sequential DYSC proposals include rejection sampling to ensure

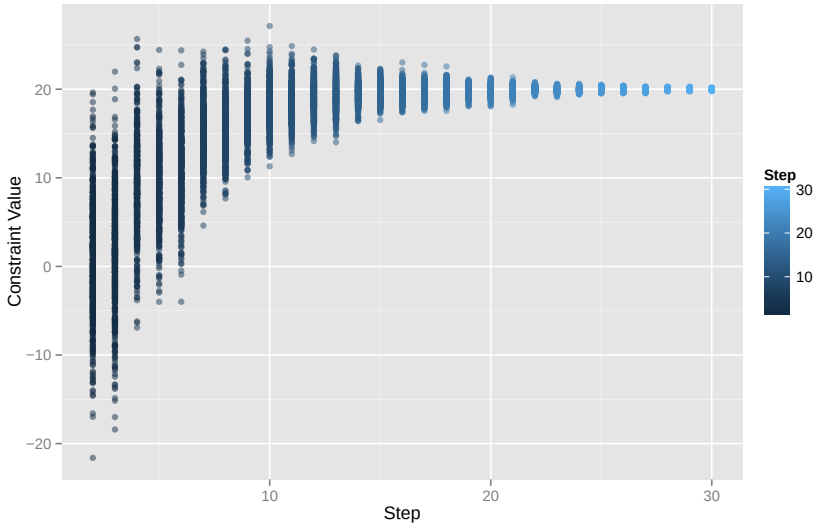

Figure 4.3: A plot demonstrating the trajectory of $f(X)$ for a 500 particle, 31 step DrSMC algorithm applied to the 15-dimensional multivariate normal problem described in Section 4.1.

that the sum does not exceed s at intermediate steps. Here, no such step is necessary as we are not dealing with strictly-positive variables.

Algorithm 5 : **DYSC**

for $i = 1, \ldots, N$ **do**
  $\tilde{w}_i \leftarrow 1$
  for $j = 1, \ldots, k$ **do**
    $\eta_j \leftarrow (s - \sum_{m=1}^{j-1} X_m)/(k+1-j)$
    $\bar{q}_j(\cdot) \leftarrow q_j(\cdot|\eta_j)$
    Draw $X_j \sim \bar{q}_j(\cdot)$
    $\tilde{w}_i \leftarrow \tilde{w}_i / \bar{q}_j(X_j)$
  end for
  $X^{(i)} \leftarrow (X_1, \ldots, X_k)$
  $w_i \leftarrow \tilde{w}_i \gamma(X^{(i)})$
end for
$w_{1:N} \leftarrow w_{1:N} / \sum_{i=1}^{N} w_i$

Consider running Algorithm 5 with $N = 250000$ particles to approximate the

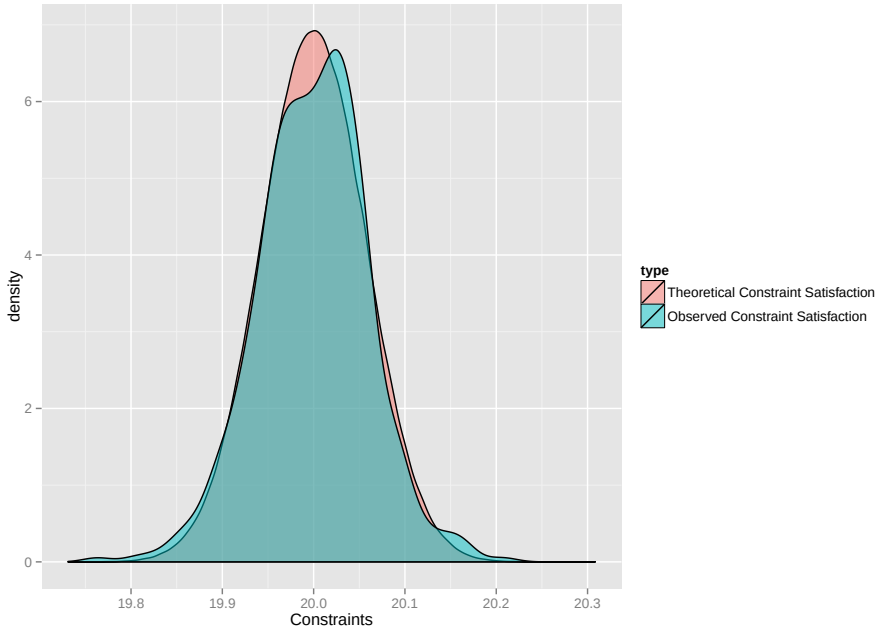

Figure 4.4: A density plot (shown in green) of $f(X)$ at the 30th step (the final step before the exact enforcement step) of a 500 particle DrSMC algorithm applied to the 15-dimensional multivariate normal problem described in Section 4.1. The theoretical distribution for $f(X)$ at this step is shown in red.

synthetic target distribution described by Section 4.1. This value of N was chosen to achieve a computation time of five minutes per run. Figure 4.5 illustrates an overlay of the DYSC posterior estimates and the true posterior distributions. It is a DYSC analog to Figure 4.1 for DrSMC. Comparing Figure 4.5 to Figure 4.1, DrSMC clearly outperforms DYSC on our synthetic problem. The DYSC approximation fails to capture the unimodal structure of the posterior. This failure is primarily due to the high variability in the importance weights which results in few particles accounting for the majority of the the normalized weight. Seeing as the importance distribution in DYSC did not capture the correlated structure of the prior, this was not unexpected. Further comparisons of DYSC and DrSMC, as well as SCMC, are available in Section 4.3.3.

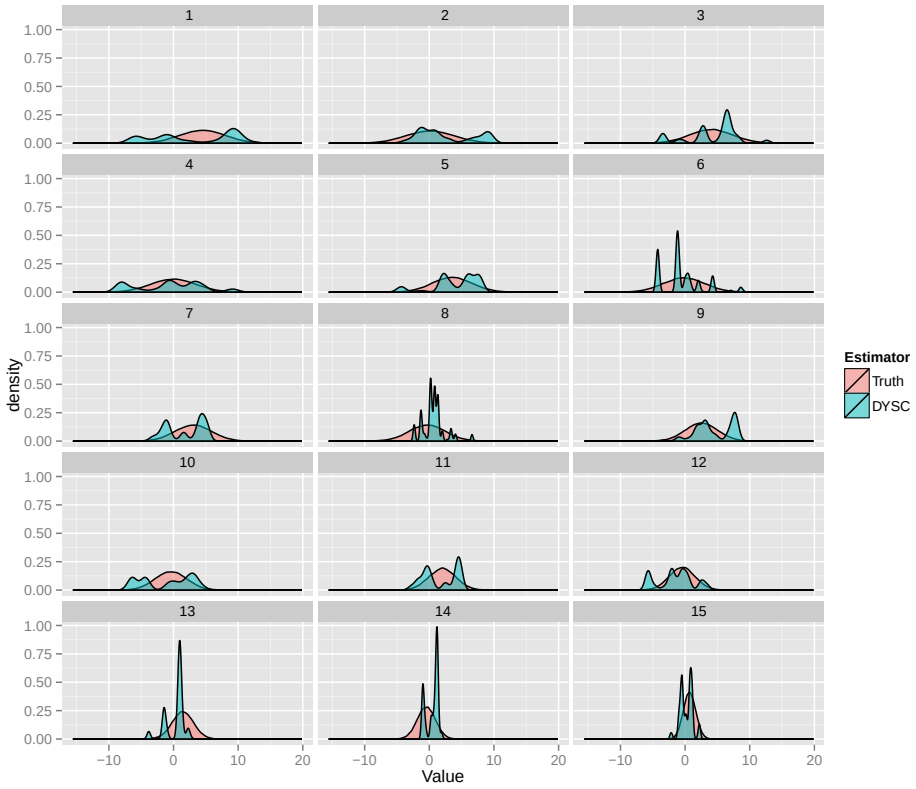

Figure 4.5: A plot overlaying the DYSC posterior density estimates for 250000 particles (shown in red) and the true theoretical values (shown in green) for the 15-dimensional multivariate normal problem described in Section 4.1

4.3.2 SCMC

Sequentially constrained Monte Carlo (SCMC) [17] is a recently developed class of SMCS for sampling from distributions which possess constraints. SCMC is a general framework capable of enforcing a wide variety of constraints, including the deterministic constraints considered in this thesis. Like DrSMC, SCMC targets a constrained distribution by initializing particles from an unconstrained distribution then propagating them through a series of intermediate distributions to gradually enforce the constraint. For constraints such as the one described in Section 4.1, [17] recommend that the series of intermediate distributions follow the form which

Table 4.2: Configuration of the SCMC with MH proposals used to approximate the distribution described in Section 4.1.

SMCS Parameters	
Number of Particles	3500
Resample Threshold	1750
Number of Steps	100
Density Parameter	
$\tau(i)$	$\exp(i-1)$
Proposal Parameter (sd)	
$\sigma(i)$	i^{-1}

was described in Section 3.2.1.

We follow this structure when specifying a SCMC algorithm for approximating the posterior distribution described in Section 4.1. Table 4.2 summarizes the parameters of SCMC used in the approximation. Each proposal is a random walk MH step in a randomly selected dimension. In order to maintain a reasonable MH acceptance rate, the standard deviation of the random walk decreases as the inverse of the step number. A similar strategy is employed for the intermediate distributions. In order to maintain a reasonable effective sample decline, the parameter τ grows exponentially in the step size. Figure 4.6 illustrates an overlay of the SCMC posterior estimates and the true posterior distributions (an analog of Figures 4.1 and 4.5). It is visually evident by comparing Figure 4.6 to Figure 4.1 that SCMC does not perform as well as DrSMC. Further comparisons are available in Section 4.3.3.

4.3.3 Comparison of Algorithms

The results of Sections 4.2-4.3.2 suggest that DrSMC markedly outperforms both DYSC and SCMC on the problem being considered. Although these illustrations are illuminating and compelling, they are based on a single run of each algorithm. This is not enough data to make a conclusion as to the optimal algorithm for this problem. In this section, five independent runs of each algorithm are considered to verify that DrSMC performs better than its counterparts. Each algorithm's performance is assessed based on its ability to approximate the true means for each of the fifteen variables.

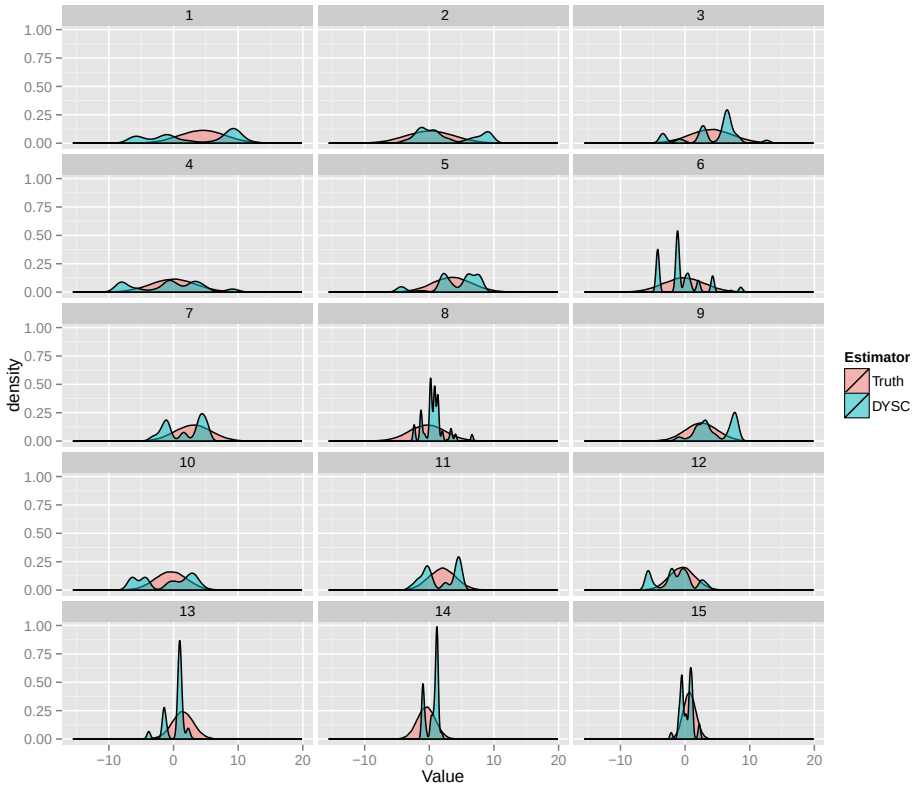

Figure 4.6: A plot overlaying the SCMC posterior density estimates for 3500 particles (shown in red) and the true theoretical values (shown in green) for the 15-dimensional multivariate normal problem described in Section 4.1.

Figures 4.7 and 4.8 illustrate the accuracy of the estimates for $\{X_1, X_3, \ldots, X_{15}\}$ and $\{X_2, X_4, \ldots, X_{14}\}$ respectively. The split of the variables into evens and odds (i.e. the two clusters), as well as the dotplot binning (to avoid point overlap), is performed to facilitate visualization. The true mean value for each variable is shown (yellow), as well as the five estimates associated with DrSMC (green), DYSC (red), and SCMC (blue). These two plots demonstrate that across multiple trials, DrSMC consistently performs at least as well as (usually better than) DYSC and SCMC.

Figure 4.9 further clarifies this point by showing the mean squared errors (MSE) in mean estimation across trials. The performance of DrSMC dominates that of

DYSC, and four of five of the DrSMC runs outperform all of the SCMC runs. Employing a Welch Two Sample t-test results in a significant difference (at an $\alpha = 0.05$ significance level) between the mean MSE for DrSMC and the mean MSE for both SCMC and DYSC.

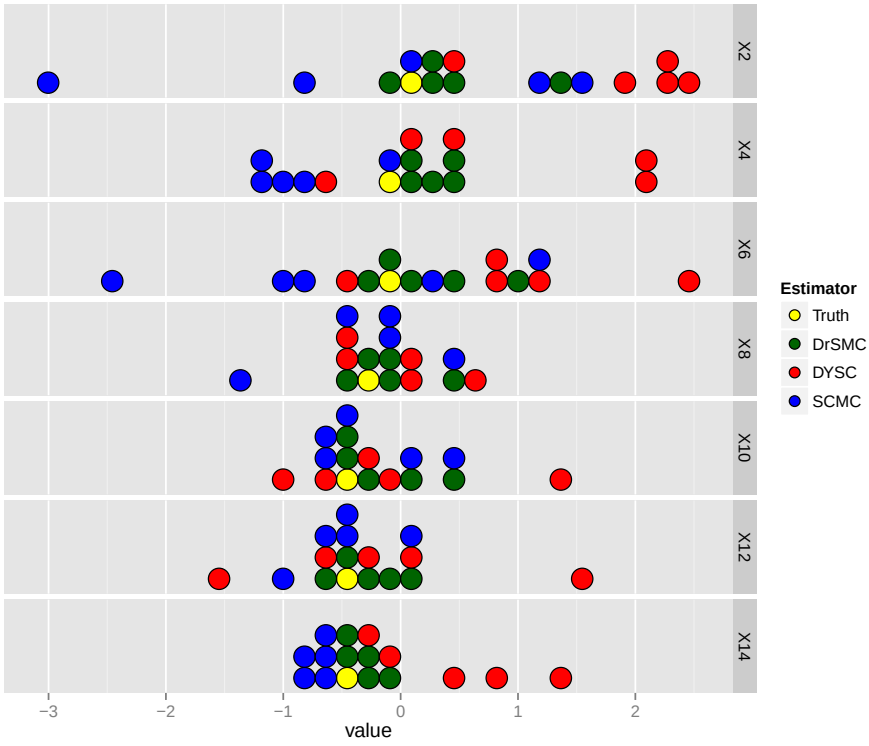

Figure 4.7: A dot plot illustrating the estimated means (odd-indexed variables) for the multivariate normal problem described in Section 4.1. The estimates are DrSMC (green), DYSC (red), and SCMC (blue), respectively. The true means (obtained analytically) are shown in yellow.

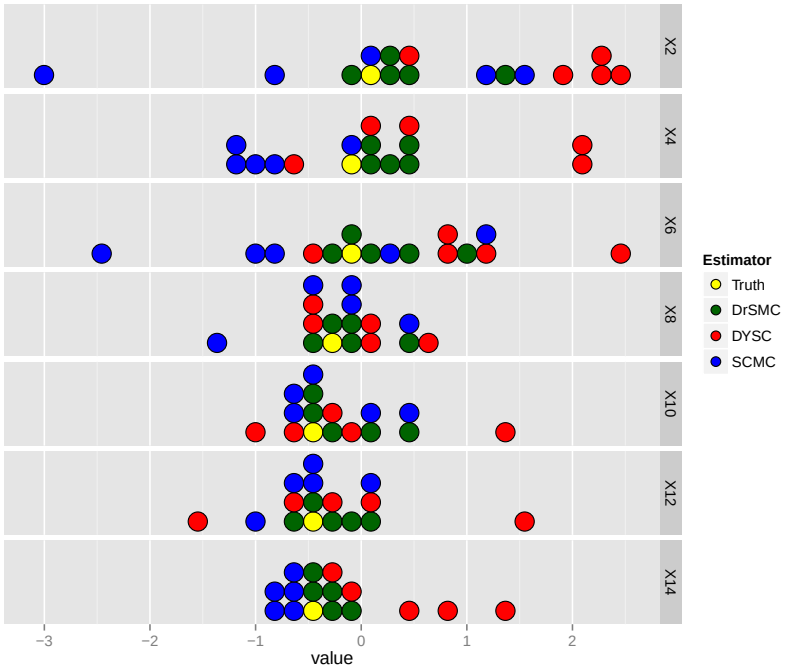

Figure 4.8: A dot plot illustrating the estimated means (even-indexed variables) for the multivariate normal problem described in Section 4.1. The estimates are DrSMC (green), DYSC (red), and SCMC (blue), respectively. The true means (obtained analytically) are shown in yellow.

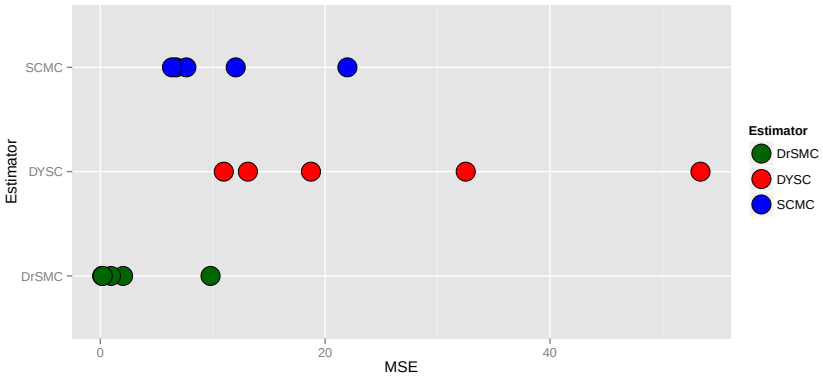

Figure 4.9: A plot demonstrating the the mean squared error for the posterior means for five runs of DrSMC (green), DYSC (red), and SCMC (blue) on problem described in Section 4.1.

Chapter 5

Conclusion

In this chapter, I make some brief concluding remarks regarding the contents of this thesis. The remarks are organized into two sections. Section 5.1 highlights some of the novel contributions in this thesis. Section 5.2 discusses some potential directions for future work in the area of DrSMC.

5.1 Novel Contributions

This thesis made several new methodological contributions on the way to developing the DrSMC algorithm and the split HMC proposal kernels for deterministic sums. Here, I briefly summarize some of the substantial contributions.

HMC within SMCS: The existing literature is sparse on using HMC to build proposals within SIS or SMC. HMC within SMC is employed in [6, 25], and HMC for annealed importance sampling is employed [28]. As far as I know, this thesis contains the first general development of HMC within a SMCS framework. Note that it is possible to formulate HMC within SMCS differently than in this thesis. For example, the algorithm could be altered such that the momentum variables are not discarded at the end of each step.

The derivation of a new Split-HMC kernel: In [26], the only types of Hamiltonian splits considered are those which exploit exact simulation. In this thesis, I

developed a new splitting scheme which exploited something other than the normal distribution (although similar to a normal distribution, the analog to the covariance matrix was not positive-definite). I derived an analytic solution to this new type of split, which resulted in a new form of split-HMC. This new kernel eliminated the problem of the vanishing epsilon for HMC kernels in deterministic sum DrSMC applications. This development creates an opportunity for HMC to be used in applications for which it was previously intractable.

Normal density formulation of constraint annealing: The general statement of DrSMC and the development of SCMC for deterministic constraints developed in [17] share many similarities. A crucial difference, however, is the functional form used to anneal in the constraints. The use of a differentiable normal density facilitates the use of HMC proposal kernels and the novel development of a suitable annealing schedule. It also allowed the development of a tractable exact enforcement step for deterministic sums.

5.2 Future Work

In the future, there are several areas that could be explored to improve the performance and/or extend the applicability of DrSMC. In this section, I propose some prospective areas for future developments which could further improve the performance of DrSMC.

HMC proposal kernel improvements: There are several possible avenues to explore for improving the proposal kernel for deterministic sums, as well as to extend split-HMC kernels to other constraints. For example, [2] developed a new split-HMC strategy called an exponential integrator. It is possible that elements of this integrator could be used to improve the split-HMC proposal kernels. Another avenue to explore is the use of different distributions for the momentum variables. Currently, a standard multivariate normal is used. However, multivariate normals with inhomogeneous variances could be useful when X possesses marginal variances which differ by orders of magnitude. There may also be a benefit to sequentially scaling the momentum variable distribution throughout the DrSMC al-

gorithm. Since DrSMC is the first use of HMC within SMCS, it represents an opportunity to develop new methods for tuning the HMC ε and L in sequential frameworks. As noted in the body of the thesis, further investigation is required into the volatility of the HMC acceptance rate at latter stages of the DrSMC algorithm.

Extending the applicability of DrSMC: Because of the many benefits specialized split-HMC proposal kernel, DrSMC performs well when the deterministic constraint is a sum. An extension of the split-HMC proposal kernel to a broader class of constraints would effectively extend the applicability of DrSMC. Such an extension would likely involve developing exact simulation of HMC for a broader class of distributions, or a clever re-parameterization of constraints which transforms them into sums. Similarly, it would be beneficial to extend the final exact enforcement step to situations other than sums. Another avenue to explore is an effective tuning algorithm for simultaneously introducing multiple components of the target density, such as two constraints, or a constraint and a likelihood. When introducing a likelihood, it may be worth investigating the use of a stochastic gradient for the HMC proposal kernel, as in [4]. Although there have been criticisms of use of stochastic gradients within an MCMC framework [1], it is worth investigating whether these criticisms extend to a SMCS framework.

Finally, due to the success of the split-HMC kernel, it is worth investigating its use for other sequential annealing approaches, such as approximate Bayesian computation [10] or simulated annealing.

Bibliography

[1] M. Betancourt. The fundamental incompatibility of scalable Hamiltonian Monte Carlo and naive data subsampling. In *International Conference on Machine Learning (ICML)*, volume (In Press), 2015. → pages 41

[2] W. Chao, J. Solomon, D. Michels, and F. Sha. Exponential integration for Hamiltonian Monte Carlo. In *International Conference on Machine Learning (ICML)*, volume (In Press), 2015. → pages 40

[3] S. Chatterjee, P. Diaconis, A. Sly, et al. Random graphs with a given degree sequence. *The Annals of Applied Probability*, 21(4):1400–1435, 2011. → pages 2

[4] T. Chen, E. Fox, and C. Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proc. International Conference on Machine Learning*, June 2014. → pages 41

[5] H. L. Chin and G. F. Cooper. Bayesian belief network inference using simulation. In *UAI*, pages 129–148, 1987. → pages 1, 2

[6] K. Choo and D. J. Fleet. People tracking using hybrid Monte Carlo filtering. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 321–328. IEEE, 2001. → pages 20, 21, 39

[7] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002. → pages 16

[8] M. Creutz. Global Monte Carlo algorithms for many-fermion systems. *Physical Review D*, 38(4):1228, 1988. → pages 10

[9] P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006. → pages 2, 4, 7, 9, 18, 24, 25

[10] P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012. → pages 41

[11] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12: 656–704, 2009. → pages 6

[12] S. Ermon, C. Gomes, A. Sabharwal, and B. Selman. A flat histogram method for inference with probabilistic and deterministic constraints. In *NIPS Workshop on Monte Carlo Methods for Modern Applications*, 2010. → pages 2

[13] J. Felsenstein. *Inferring phylogenies*, volume 2. Sinauer Associates Sunderland, 2004. → pages 1

[14] M. Fishelson and D. Geiger. Optimizing exact genetic linkage computations. *Journal of Computational Biology*, 11(2-3):263–275, 2004. → pages 1

[15] V. Gogate and R. Dechter. Samplesearch: Importance sampling in presence of determinism. *Artificial Intelligence*, 175(2):694–729, 2011. → pages 2

[16] V. G. Gogate. *Sampling Algorithms for Probabilistic Graphical Models with Determinism DISSERTATION*. PhD thesis, University of California, Irvine, 2009. → pages 1

[17] S. Golchi and D. A. Campbell. Sequentially constrained Monte Carlo. *arXiv preprint arXiv:1410.8209v2*, 2014. → pages 2, 3, 16, 17, 27, 34, 40

[18] O. Gries and R. Möller. Gibbs sampling in probabilistic description logics with deterministic dependencies. In *UniDL*. Citeseer, 2010. → pages 2

[19] M. D. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014. → pages 11, 14, 21

[20] L. Li, B. Ramsundar, and S. Russell. Dynamic scaled sampling for deterministic constraints. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pages 397–405, 2013. → pages 1, 2, 3, 27, 31

[21] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008. → pages 1, 5, 6

[22] R. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011. → pages 2, 10, 12, 14

[23] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11 (2):125–139, 2001. → pages 7, 16, 18

[24] A. Pakman and L. Paninski. Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014. → pages 12, 22

[25] E. Poon and D. J. Fleet. Hybrid Monte Carlo filtering: Edge-based people tracking. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 151–158. IEEE, 2002. → pages 20, 39

[26] B. Shahbaba, S. Lan, W. O. Johnson, and R. M. Neal. Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24(3):339–349, 2014. → pages 14, 22, 23, 39, 45

[27] M. A. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, and G. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of information in Medicine*, 30(4):241–255, 1991. → pages 1

[28] J. Sohl-Dickstein and B. J. Culpepper. Hamiltonian annealed importance sampling for partition function estimation. *arXiv preprint arXiv:1205.1925*, 2012. → pages 20, 21, 39

[29] Stan Development Team. *Stan Modeling Language Users Guide and Reference Manual, Version 2.5.0*, 2014. URL http://mc-stan.org/. → pages 2

[30] L. Tabourier, C. Roth, and J.-P. Cointet. Generating constrained random graphs using multiple edge switches. *Journal of Experimental Algorithmics (JEA)*, 16:1–7, 2011. → pages 2

[31] D. Venugopal and V. Gogate. Giss: Combining Gibbs sampling and samplesearch for inference in mixed probabilistic and deterministic graphical models. In *AAAI*, 2013. → pages 1

[32] Z. Wang, S. Mohamed, and N. de Freitas. Adaptive Hamiltonian and Riemann manifold Monte Carlo samplers. In *International Conference on Machine Learning (ICML)*, pages 1462–1470, 2013. → pages 14, 21, 23

Appendix A

Supporting Derivations

A.1 Analytically Integrating Hamiltonian Dynamics of the Random Lagrangian

In Section 3.4, I discuss separating the energy function underlying DrSMC for sum constraints into two distinct summands, $H_1(X^t_{1:k})$ and $H_2(X^t_{1:k}, S^t_{1:k})$, with H_2 possessing the following form:

$$H_2(X^t_{1:k}, S^t_{1:k}) = \frac{\left(\sum_{j=1}^{D} X^t_j - s\right)^2}{2b^2} + \frac{\sum_{j=1}^{D}\left(S^t_j\right)^2}{2}.$$

This division of the energy function facilitates the use of a split Hamiltonian integrator [26], with H_2 being integrated analytically. Here, I provide the mathematical derivation for the analytic integration.

Recall that Hamiltonian motion is governed by the following system of partial differential equations,

$$\frac{\partial H(X^t_{1:k}, S^t_{1:k})}{\partial X_i} = -\frac{\mathrm{d}S^t_i}{\mathrm{d}t},$$
$$\frac{\partial H(X^t_{1:k}, S^t_{1:k})}{\partial S^t_i} = \frac{\mathrm{d}X^t_i}{\mathrm{d}t}.$$

Differentiating H_2 yields

$$\frac{\partial H(X^t_{1:k}, S^t_{1:k})}{\partial X^t_i} = \frac{\sum_{j=1}^{D} X^t_j - s}{b^2},$$

$$\frac{\partial H(X^t_{1:k}, S^t_{1:k})}{\partial S^t_i} = S^t_i.$$

It follows that

$$\frac{\mathrm{d}^2 X^t_i}{\mathrm{d}t^2} = -\frac{\sum_{j=1}^{D} X^t_j - s}{b^2},$$

which in turn implies that

$$X^t_i = X^t_1 + (S^0_i - S^0_1)t + (X^0_i - X^0_1).$$

Therefore, our system can be expressed as a set of independent equations of the form

$$\frac{\mathrm{d}^2 X^t_i}{\mathrm{d}t^2} = -\frac{DX^t_i + \alpha_i t + \beta_i - s}{b^2}$$

for constants α_i and β_i where

$$\alpha_i = \sum_{j=1}^{D} S^0_j - DS^0_i,$$

$$\beta_i = \sum_{j=1}^{D} X^0_j - DX^0_i.$$

The solution to such an equation yields an expression for X is of the form:

$$X^t_i = k_1 \cos(\sqrt{D}t/b) + k_2 \sin(\sqrt{D}t/b) - \frac{\alpha_i t + \beta_i - s}{D}.$$

Differentiating with respect to t yields an expression for the momentum

$$S^t_i = -\frac{\sqrt{D}}{b} k_1 \sin(\sqrt{D}t/b) + \frac{\sqrt{D}}{b} k_2 \cos(\sqrt{D}t/b) - \frac{\alpha_i}{D}.$$

Expressions for k_1 and k_2 can be derived from the initial conditions as follows.

$$X_i^0 = k_1 - (\beta_i - s)/D \Rightarrow k_1 = X_i^0 + (\beta_i - s)/D.$$

$$S_i^0 = \frac{\sqrt{D}}{b} k_2 - \alpha_i/\sqrt{D} \Rightarrow k_2 = \frac{b}{\sqrt{D}} \left(S_i^0 + \alpha_i/D\right).$$

Using these expressions, $X_{1:k}^t$ and $S_{1:n}^t$ depend on the initial conditions $X_{1:k}^0$ and $S_{1:n}^0$ in the following manner:

$$X_i^t = \left(\bar{X}^0 - s/D\right)\cos(\sqrt{D}t/b) + \frac{b\bar{S}^0}{\sqrt{D}}\sin(\sqrt{D}t/b) - (\bar{S}^0 - S_i^0)t - (\bar{X}^0 - X_i^0) + \frac{s}{D},$$

$$S_i^t = -\frac{\sqrt{D}\left(\bar{X}^0 - C/D\right)\sin(\sqrt{D}t/b)}{b} + \bar{S}^0\cos(\sqrt{D}t/b) - (\bar{S}^0 - S_i^0),$$

where $\bar{X}^0 = \sum_{j=1} X_j^0/D$ and $\bar{S}^0 = \sum_{j=1} S_j^0/D$.

A.2 Functional Form for DrSMC Intermediate Distributions

In SMCS, it is crucial for the sequence of intermediate distributions to introduce change gradually. Otherwise, the sampler can quickly devolve into a state of "particle degeneracy", in which essentially all of the weight is assigned to a single particle. Propagating forward, all final particles are descendants of this same ancestor, resulting in poor approximation of the target distribution. Degeneracy can be avoided by controlling the effective sample size (ESS) between steps. This section provides an approach for controlling ESS in DrSMC.

In DrSMC, the dynamics of the intermediate distributions are driven by the sequence of constraint satisfaction variances (annealing schedule), given by $b_{1:p}$. This sequence is the mechanism through which changes in ESS can be controlled. Recall that the normal variance of the relationship violation at step n is b_{n-1}, and the weight update at is given by $\tilde{w}(X_{n-1},X_n) = \gamma_n(X_{n-1})/\gamma_{n-1}(X_{n-1})$. In the absence of any likelihood annealing, this weight update reduces to

$$\tilde{w}(X_{n-1},X_n) = \frac{b_{n-1}}{b_n}\exp\left(-\frac{(f(X_{n-1})-s)^2}{2b_n^2}+\frac{(f(X_{n-1})-s)^2}{2b_{n-1}^2}\right).$$

To prevent rapid ESS drops, I propose a functional form of the sequence which explicitly controls the weight updates of particles, proportional to the degree to which they violate the deterministic relationship $f(X) = s$. Specifically, b_n is defined such that at step n, the weight update of a particle X' which perfectly satisfies the relationship is a factor of C greater than that of a particle X which is b_{n-1} (one standard deviation) away from satisfying the relationship (for some $0 < C < 1$).

Without loss of generality, suppose $f(X') = s$ and $f(X) = s + b_{n-1}$. The ratio of the weight update for X and the weight update for X' at step n is given by:

$$\frac{\tilde{w}(X'_{n-1},X'_n)}{\tilde{w}(X_{n-1},X_n)} = \exp\left(\frac{1}{2}\left(\frac{b_{n-1}^2}{b_n^2}-1\right)\right)$$

Setting this equal to C yields the relationship

$$b_n = b_{n-1}(1+2\log(C))^{-1/2}$$

which justifies a geometric decay for $b_{1:p}$. With this derivation, tuning $b_{1:p}$ becomes relatively painless. The only two parameters to tune are b_1 and C.

A.3 Exact Distribution of Multivariate Normal given its Sum

Consider an k-dimensional multivariate random normal distribution $X \sim N(\mu, \Sigma)$. Now consider applying the transformation $Y_i = X_i$ for $i = 1, \ldots, k-1$ and $Y_k = X_1 + X_2 + \cdots + X_k$. This transformation can be expressed as $Y = AX$, where

$$A = \begin{bmatrix} I_{k-1} & 0_{k-1} \\ 1_{k-1}^T & 1 \end{bmatrix}.$$

Here, I_{k-1} denotes a $(k-1) \times (k-1)$ identity matrix, 0_{k-1} is a column vector of $k-1$ zeroes and 1_{k-1}^T is a row vector of $k-1$ ones. Since the transformation is linear, it follows from the properties of the multivariate normal that $Y \sim N(A\mu, A\Sigma A^T)$. Conditioning on $Y_k = s$, the joint distribution of $Y_{1:(k-1)} = X_{1:(k-1)}$ is given by $N(\mu', \Sigma')$, where $\mu' = \mu + \Sigma_{1:(k-1),k}(s - \sum_{i=1}^{k} \mu_i)/\Sigma_{k,k}$ and $\Sigma' = \Sigma_{1:(k-1),1:(k-1)} - \Sigma_{1:(k-1),k}\Sigma_{k,1:(k-1)}/\Sigma_{k,k}$. Finally, $X_k = s$ is then specified by the condition.