Adaptive Monte Carlo Integration

by

James Neufeld

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Monte Carlo methods are a simple, effective, and widely deployed way of approximating integrals that prove too challenging for deterministic approaches. This thesis presents a number of contributions to the field of adaptive Monte Carlo methods. That is, approaches that automatically adjust the behaviour of the sampling algorithm to better suit the targeted integrand. The first of such contributions is the introduction of a new method, antithetic Markov chain sampling, which improves sampling efficiency through the use of simulated Markov chains. These chains effectively guide the sampling toward more influential regions of the integrand (modes). We demonstrate that this approach leads to unbiased estimators and offers significant improvements over standard approaches on challenging multi-modal integrands.

We next consider the complementary task of efficiently allocating computation between a set of unbiased samplers through observations of their past performance. Here, we show that this problem is equivalent to the well known stochastic multi-armed bandit problem and, as a result, existing algorithms and theoretical guarantees transfer immediately which gives rise to new results for the adaptive Monte Carlo setting. We then extend this framework to cover an important practical condition, where each individual sampler (bandit arm) may take a random amount of computation time to produce a sample. Here, we again show that existing bandit algorithms can be applied through the use of a simple sampling trick, and prove new results which bounding the regret for any such algorithm from above.

Lastly, we consider the task of combining a set of unbiased Monte Carlo estimators, with unique variances and samples sizes, into a single estimator. We show that upper-confidence approaches similar to those used in the multi-armed bandit literature lead to estimators that improve on existing solutions both theoretically and in practice. Interestingly, each of these contributions may be applied in parallel and in complement to one another to produce any number of highly adaptable, robust, and practical Monte Carlo integration algorithms.

Acknowledgements

Naturally, a PhD thesis is only ever attributed to a single author. While this tradition celebrates the hard work and perseverance of the student, it unfortunately does not reflect collaborative nature of the work. As such, in this document the first person plural is used as an acknowledgement to the many individuals that made this work possible.

In particular, I want to thank my supervisors Dale Schuurmans and Michael Bowling for their technical know-how, guidance, and invaluable advice throughout my graduate studies. Also, I am exceedingly grateful to my collaborators Csaba Szepesvári and András György who's expertise, optimism, and insightful suggestions were instrumental in translating many high level ideas into concrete contributions. Thank you as well to my many friends and colleagues at the University of Alberta who made my graduate studies such a treasured experience.

Of course, thank you to my wonderful parents and sisters who helped in far too many ways to count, and my loving wife Thea and three amazing children Paige, Breanna, and Oliver who made this whole effort worthwhile.

Table of Contents

List of Figures

Table of Notation

Quantity	Notation	Comment
Random variable	X, Y	uppercase
Constant variable	n, k, m	lowercase
Sample space	$\mathcal{X}, \mathcal{Y}$	uppercase script
Probability triple	$(\mathcal{X}, \mathcal{P}, \mathcal{B})$	sample space, probability distribution, and σ-algebra
Distribution function (cdf)	$\Pi(x), \Pi(x\|\theta), G(x), \Pi_\theta(x)$	
Density function (pdf)	$\pi(x), \pi(x\|\theta), g(x), \pi_\theta(x)$	
Parameter	$\mu, \theta, \zeta, \mu(\mathcal{P})$	lowercase Greek letters, arguments omitted when clear
Parameter space	Θ, Ω	uppercase Greek letters
Estimator	$\hat{\theta}, \hat{\mu}, \hat{\mu}(X_1, ..., X_n), \hat{\mathcal{Z}}$	symbol corresponds to the parameter being estimated, arguments omitted when clear
Indication	$\mathbb{I}\{x > y\}, \mathbb{I}\{x \in \mathcal{A}\}$	evaluates to 1 if the predicate is true, 0 otherwise
Expectation	$\mathbb{E}[X], \mathbb{E}[h(X)], \mathbb{E}_\pi[X]$	
Variance	$\mathbb{V}(X), \mathbb{V}(h(X)), \mathbb{V}_\pi(X)$	
Probability	$\mathbb{P}(X > y), \mathbb{P}_\pi(X > y)$	
Definition	$f(x) := x^2$	
Sequence	$(X_1, ..., X_n), X_{1:n}$	

Chapter 1

Introduction

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin"

– John von Neumann

In this thesis we explore the computation of expected values, in particular through the use of Monte Carlo methods. Specifically, given a random variable $X \in \mathcal{X}$, distributed according to a probability measure admitting the density function π, our interest surrounds expectations expressed as:

$$\mathbb{E}_\pi \left[h(X) \right] := \int h(x)\pi(x)dx, \tag{1.1}$$

where $h : \mathcal{X} \to \mathbb{R}$ is a bounded measurable function assumed only to be evaluable at any point in the domain. Although this formulation is mathematically concise and widespread in its application, efficient computation of the solution remains a notoriously challenging task.

Indeed, the above expectation appears in a number unique and interesting settings across many disciplines in science and engineering. For instance, h might represent the price of a financial option and π our uncertainty about the underlying asset. Or h might define the payoff function for a high-stakes game of chance while π describes the likelihood of individual games states. Alternatively, h may represent the local concentration of a particular dissolved compound while π reflects the fluid dynamics for the entire solution. For any of these settings efficient computation of the integral in Eq. (1.1) is likely a necessary step in making consequential, and possibly time sensitive, decisions.

What typically makes the task of solving this integral so challenging is the fact that the exact analytical forms of h and π are not often available. As a result, the symbolic integration approaches from one's calculus textbook cannot be applied. Ultimately, for most

practical scenarios, approximating the solution numerically is the only viable approach. Numerical integration techniques generally fall into one of two categories, which we will refer to as *quadrature* methods and *Monte Carlo* methods. Importantly, we find the critical distinction between these classes is not the use of randomization, but whether some form of interpolation is attempted between evaluated points on the integrand.

The use of interpolation is a double-edged sword as it can offer enormous advantages for problems with smoothly varying h and π in lower dimensions, but is challenging to apply to non-smooth or higher dimensional integrands. At the very least, an interpolation approach must evaluate the integrand at a minimum of 2^d locations, where d is the dimensionality of $\mathcal{X}$. This exponential *sample complexity* limits the applicability of this approach when it comes to larger and more impactful problems.

Monte Carlo integration approaches, on the other hand, approximate integrals by querying the integrand at random, or quasi-random, locations and return a weighted empirical sum (average) of these points. For instance, if we are able to efficiently collect n samples, $X_1, ..., X_n$, distributed according to π, the integral in Eq. (1.1) may then be approximated the estimator

$$\hat{\mu}_n^{MC} := \frac{1}{n} \sum_{i=1}^{n} h(X_i).$$

Letting $\mu := \mathbb{E}[h(X)]$ we observe that the above estimator is unbiased: $\mathbb{E}\left[\hat{\mu}_n^{MC}\right] = \mu$ and has a mean squared error (MSE) given by $\mathbb{E}\left[\left(\hat{\mu}_n^{MC} - \mu\right)^2\right] = \frac{1}{n}\mathbb{V}(h(X))$.

This straightforward analysis establishes a convergence rate of $O(n^{-1})$ which, critically, is independent of the dimensionality of the integral. In this way, the Monte Carlo integration method can be seen to break the infamous *curse of dimensionality* that plagues quadrature methods; at least in theory. Despite this powerful theoretical result, naive Monte Carlo implementations remain highly inefficient for moderately complex problems. Further, there is no single Monte Carlo approach that can be applied effectively across disciplines and problems. For this reason, a large variety of unique approaches have been introduced in the literature each exploiting specific characteristics of the problem at hand. In this thesis we explore Monte Carlo methods that, in some form or another, alter their sampling behaviour automatically depending on the properties of the supplied integral with the intention of improving the accuracy of the final estimate. In this way, we lessen the requirement that practitioners understand, implement, and experiment with a large number of specialized techniques for any new problem.

1.1 Contributions

This thesis includes a number of interrelated and novel contributions to the field of adaptive Monte Carlo methods organized into four main chapters summarized below.

Formulation of the Antithetic Markov Chain Method

In Chapter 3 we present an extended treatment of the antithetic Markov chain sampling (AMCS) method originally presented in Neufeld et al. (2015). This approach is a combination of sequential Monte Carlo sampling methods and the method of antithetic variates. More simply, AMCS reduces approximation error by effectively searching out regions of the integrand that exhibit large changes in magnitude (peaks) through the use of simulated Markov chains. By averaging within these regions, much of the variability of the integrand can be removed, resulting in a reduced approximation error of the final estimate. The AMCS estimator is shown to be unbiased (Theorem 3.1) and expressions for the variance and sample error, considering the computational footprint, are derived. Finally, we provide explicit parameterizations for AMCS (Markov kernels and stopping rules) and empirically demonstrate their utility on non-trivial machine learning tasks that challenge existing methods.

Formulation of the *Learning to Select Monte Carlo Samplers* Framework

Chapter 4 introduces a new adaptive Monte Carlo framework in which a decision making agent is presented with a set of K unbiased Monte Carlo samplers for the same statistic (μ) and tasked with choosing which samplers to draw samples from in order to compute μ most efficiently; this work was originally published in Neufeld et al. (2014). We formulate an expression for the MSE of any given allocation policy for this sequential allocation task and define a regret formula expressing this error in relation to the optimal policy. Under this notion of regret, we prove this problem is *equivalent* to the classic stochastic multi-armed bandit problem (Theorem 4.1 and Theorem 4.2). As a direct result of this reduction, we are able to show that the regret upper bounds for many of the standard bandit approaches apply immediately to this setting (Corollary 4.1), in addition to the classic Lai-Robbins lower bound (Theorem 4.3). Lastly, we demonstrate that the existing bandit algorithms can significantly outperform existing population-based adaptive Monte Carlo methods in standard adaptive importance sampling tasks.

Extensions to Monte Carlo Samplers With Non-Uniform Costs

In the adaptive Monte Carlo framework detailed in Chapter 4 it is assumed that each of the underlying samplers require the same amount of computation to produce a single sample. However, for many of the more sophisticated Monte Carlo techniques (such as AMCS) the per-sample computational cost can be unknown, stochastic, or vary with the parameterization of the method. In order to apply the bandit-based adaptive sampling routines to these more sophisticated samplers we extend the previous formulation to account for *non-uniform* (stochastic) costs in Chapter 5. Ultimately, under mild technical assumptions, we show that through a straightforward sampling trick an $\widetilde{O}(\sqrt{t})$ regret is achievable by standard bandit approaches in this setting (Theorem 5.1). We go on to show that these techniques can be used to develop an adaptive-AMCS variant that can outperform any fixed AMCS variant on the same suite of problems used in Chapter 3. We do, however, uncover a interesting negative result (linear regret) which can occur when there is more than one optimal sampler; we show that this problem will regularly surface when selecting between standard sequential Monte Carlo sampling algorithms. We go on to show that this poor performance stems from the simplifying decision to use the empirical (unweighted) average of the sampled values as the final estimate.

An Upper-Confidence Approach to the Weighted Estimation of a Common Mean

In Chapter 6 we consider the general task of constructing a weighting for a convex combination of unbiased estimators in order to produce a single estimate that minimizes the MSE. This formulation is applicable to a number of practical settings but is uniquely useful in addressing the issues uncovered in the non-uniform cost setting mentioned above. We first demonstrate that by weighting each estimator inversely proportional to its variance, one recovers the unique minimum variance unbiased estimator as well as a minimax estimator (Theorem 6.1). Using this approach as an optimal comparison we construct a regret formulation for this task and introduce the UCB-W estimator which uses an upper-confidence estimate of the sample variance to construct a weighting. We go on to show that this estimator achieves a $\widetilde{O}(\sqrt{t})$ regret in the case where samples are selected deterministically (Theorem 6.2) or according to a random stopping rule (Theorem 6.3); thus generalizing the regret bounds proved in Chapter 5 to cover the edge-cases that resulted in linear regret. We evaluate the proposed UCB-W estimator and show that it offers, frankly, massive savings in both the uniform cost and non-uniform cost bandit settings and by far out-performs the existing Graybill-Deal estimator which weights each estimator using its sample variance.

Chapter 2

Background

"Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse"

– Alan D. Sokal

Monte Carlo integration approches represent a uniquely powerful class of algorithm yet, at the same time, the algorithms are typically straightforward to understand, implement, and extend. Indeed, the ease at which problem-specific extensions are constructed has resulted in a number of unique Monte Carlo algorithms represented in the literature. In this chapter, we provide a high level overview of the more popular approaches from which many of the more sophisticated methods are based.

An important precursor to understanding the tradeoffs between different Monte Carlo integration approaches is to identify a means by which to assess the quality of a given estimator. For any distribution $\Pi \in \mathcal{P}$ belonging to the probability triple $(\mathcal{X}, \mathcal{P}, \mathcal{B})$, parameter space Θ, and a parameter of interest $\mu : \mathcal{P} \to \Theta$, we define an estimator as a mapping $\hat{\mu} : \mathcal{X}^n \to \Theta$. Note that for the purposes of this thesis we will consider the sample space to be $\mathcal{X} = \mathbb{R}^d$, $\mathcal{B}$ to be the Borel σ-algebra, and $\mathcal{P}$ to be the Lebesgue measure unless otherwise specified. Additionally, in the sequel we make use of the shorthand $\hat{\mu}_n := \hat{\mu}_n(X_1, ..., X_n)$, with $(X_1, ..., X_n) \sim \Pi$, and $\mu := \mu(\Pi)$ when the arguments are clear from the available context. In this notation, we can evaluate the quality of an estimator using a loss function $L(\hat{\mu}_n, \mu)$ and define the *risk* of the estimator as the expected loss: $R(\hat{\mu}_n, \mu) := \mathbb{E}[L(\hat{\mu}_n, \mu)]$. While there are a variety of different loss functions and parameter spaces considered in the literature, in this thesis we focus on the most common setting where $\Theta = \mathbb{R}$ and L is the L2 loss function: $L(x, y) = (x - y)^2$. Here, the risk reduces to the well known *mean squared error* (MSE) measure, denoted as $\mathrm{MSE}(\hat{\mu}_n, \mu) := \mathbb{E}\left[(\hat{\mu}_n - \mu)^2\right]$.

An important property of the MSE is the bias-variance decomposition $\mathrm{MSE}(\hat{\mu}_n) = \mathbb{E}[\hat{\mu}_n - \mu]^2 + \mathbb{E}\left[(\hat{\mu}_n - \mathbb{E}[\hat{\mu}_n])^2\right] =: \mathrm{bias}(\hat{\mu}_n, \mu)^2 + \mathbb{V}(\hat{\mu}_n)$, which gives rise to the common convergence measures used in the evaluation of any Monte Carlo method, *unbiasedness*, when $\mathbb{E}[\mu_n] = \mu$, and *consistency*, when $\lim_{n\to\infty} \mathbb{P}(|\mu_n - \mu| \geq \varepsilon) = 0$. In the context of Monte Carlo samplers, unbiasedness is generally considered the "stronger" of the two conditions as it often implies the latter. For instance, in the usual setting, where $(X_1, ..., X_n)$ are i.i.d. distributed, the basic MC estimator μ_n^{MC} is unbiased and has MSE given by $\frac{1}{n}\mathbb{V}(X) = O(n^{-1})$, which can be used (together with Chebyshev's inequality) to establish consistency.

2.1 Variance Reduction

Many of the more popular Monte Carlo procedures use a sequence of i.i.d. samples and lead to unbiased estimators. Therefore, techniques for reducing the approximation error typically involve reducing the variance of individual samples and, as a result, many of the classic computation-saving Monte Carlo techniques are referred to as *variance reduction* methods. In this section we review some of these techniques as they are the foundation for many of the more powerful approaches we will later consider.

2.1.1 Importance Sampling

In many cases approximation of the integral in Eq. (1.1) by directly sampling from the *target distribution* (π) is computationally infeasible. This occurs either due to the fact that generating samples is too expensive or because π is poorly matched to h and individual samples are highly variable. The latter case typically results when π assigns low probability to regions where the target function h has large magnitude. Importance sampling is one of the standard ways of addressing these issues. The approach is straightforward; one generates samples from a *proposal distribution* π_0 that is inexpensive to simulate and, ideally, assigns high likelihood to the high magnitude or *important* regions of h. The sampling bias introduced by this change of measure is removed through the application of an *importance weight*. Specifically, given a sequence of n i.i.d. random variables $(X_1, ..., X_n)$ distributed according to π_0, the importance sampling estimator is given by

$$\hat{\mu}_n^{IS} := \frac{1}{n}\sum_{i=1}^{n} w(X_i)h(X_i), \tag{2.1}$$

where $w(x) = \frac{\pi(x)}{\pi_0(x)}$ defines the importance weighting function.

The above estimator is unbiased and consistent provided that $\text{supp}(\pi) \subseteq \text{supp}(\pi_0)$ and $\mathbb{V}_{\pi_0}(w(X)) < \infty$. Additionally, the method will result in a lower MSE whenever $\mathbb{V}_{\pi_0}(w(X)h(X)) < \mathbb{V}_{\pi}(h(X))$, indeed for non-negative h when the proposal density is proportional to the integrand, $\pi_0 = \pi^* := h\pi / \int h(x)\pi(x)dx$, we have $\mathbb{V}_{\pi_0}(w(X)h(X)) = 0$. Unfortunately, using this proposal is not possible in practice since the necessary normalizing constant for π^* is equal to the original unknown (μ). Though, this identity does give the practitioner an indication as to how to select an effective proposal.

In some cases, the target distribution π can be evaluated only up to an unknown normalizing constant ζ, which often must be approximated in order to get an accurate estimate of μ. Also, in some settings the ζ quantity is of independent interest, for example the classic task of Bayesian model comparison (see Robert (2010)). An interesting advantage of importance sampling is that it can be used to approximate both μ and ζ from the same set of samples. Specifically, if we denote the un-normalized density as $\hat{\pi} := \frac{\pi}{\zeta}$ we can use the unbiased estimator

$$\hat{\zeta}_n^{IS} := \frac{1}{n}\sum_{i=1}^{n} w(X_i), \tag{2.2}$$

with $w(x) = \frac{\hat{\pi}(x)}{\pi_0(x)}$. Using this estimate we can then approximate μ with the *weighted importance sampling* estimator

$$\hat{\mu}_n^{WIS} := \frac{\frac{1}{n}\sum_{i=1}^{n} w(X_i)h(X_i)}{\hat{\zeta}_n^{IS}}. \tag{2.3}$$

As a result of the division by the random quantity $\hat{\zeta}$, this estimator cannot be said to be unbiased. However, the bias can be shown to decrease at a rate of $O(n^{-1})$ (Powell and Swann, 1966). Also, even in cases where the normalizing constant is known, the weighted importance sampling estimator will often outperform the unbiased version.

A primary challenge with deploying importance sampling is constructing proposals that assign sufficient density to the tail regions of π. If, for example, the tails of the proposal converge to zero faster than the target distribution, then the importance weights will diverge and $\mathbb{V}(w(X))$ will not be finite. Moreover, because these problematic regions are never sampled in practice, it is effectively impossible to diagnose the problem numerically; the unsuspecting practitioner will observe consistent, steady, convergence to the same incorrect estimate even between multiple runs. To address this difficulty, it is common to employ a so-called *defensive sampling* strategy where a heavy-tailed distribution, such as a multivariate Student density, is added to π_0 with a small mixing coefficient (see Robert and Casella (2005)). However, while this approach generally ensures the variance will be finite,

in practice it often does little to ensure the variance will not be prohibitively high. Consequently, engineering proposal densities and numerically diagnosing the convergence of an IS estimate, especially in high dimensions, remains a serious practical concern. It is worth noting also that this problem is often exacerbated by attempts to *adapt* the proposal to the integrand using previously sampled points, as we explain in Section 2.2.

2.1.2 Antithetic Variates

One of the most straightforward variance reduction techniques is the *method of antithetic variates*. The approach works in the following way: suppose we have two sets of correlated random variables, $(X_1, ..., X_n)$ and $(Y_1, ..., Y_m)$, such that $\mathbb{E}[X] = \mathbb{E}[Y] = \mu$, $X_i \perp\!\!\!\perp X_j$, $Y_i \perp\!\!\!\perp Y_j$, and $X_i \not\perp\!\!\!\perp Y_i$ then the estimator

$$\hat{\mu}_n^{AV} := \frac{1}{n} \sum_{i=1}^{n} \frac{X_i + Y_i}{2}$$

is unbiased and has variance given by $\mathbb{V}\left(\hat{\mu}_n^{AV}\right) = \frac{1}{4n}\mathbb{V}(X + Y) = \frac{1}{4n}(\mathbb{V}(X) + \mathbb{V}(Y) + \mathrm{Cov}(X, Y))$, where it is understood that $X \equiv X_i$ and $Y \equiv Y_i$ for any i. This implies that the estimator $\hat{\mu}_n^{AV}$ will offer a reduction over the vanilla Monte Carlo estimator (using $2n$ samples) whenever X and Y are negatively correlated.

For example, suppose that $X \sim \mathrm{Uniform}(a, b)$ and that h is a monotonically increasing or decreasing function. If we define the antithetic variate as $Y = b + a - X$ it is clear that $\mathbb{E}[h(Y)] = \mathbb{E}[h(X)] = \mu$ and since h a monotonic function, it is straightforward to establish $\mathrm{Cov}(h(X), h(Y)) < 0$. However, as with importance sampling, practical scenarios where the practitioner has enough *a priori* knowledge to design variance-reducing antithetic transforms are not as common as one might like.

2.1.3 Stratified Sampling

Stratified sampling is a variance reduction technique that is used frequently in classic statistics settings such as opinion polling or population estimation. In such cases, the sample population is often endowed with a natural partition based on characteristics such age, sex, ethnicity, etc. The general idea of this approach is to approximate an average for each of the subsets separately, paying more attention to the difficult ones, and combine these estimates in proportion to their respective sizes.

Suppose our sample space $\mathcal{X}$ decomposes into a partition $(\mathcal{X}_1, ..., \mathcal{X}_k)$ that permits sep-

arate Monte Carlo approximations of the integral on each domain; that is,

$$\int_{\mathcal{X}} h(x)\pi(x)dx = \sum_{i=1}^{k} \lambda_i \int_{\mathcal{X}_i} h(x)\pi_i(x) = \sum_{i=1}^{k} \lambda_i \mu_i,$$

where λ_i gives the probability of each region (strata) under π and π_i denotes the density proportional to π within this region. As with π, it is required that each π_i permit efficient point-wise evaluation. If we let $\hat{\mu}_{i,n}$ denote an unbiased Monte Carlo estimator for strata μ_i at time[1] n then the stratified sampling estimator is given by

$$\hat{\mu}_n^{SS} = \sum_{i=1}^{k} \lambda_i \hat{\mu}_{i,n}.$$

This estimator is unbiased and, assuming each estimator uses n_i independent samples (chosen deterministically with $n = \sum_{i=1}^{k} n_i$) with variance σ_i^2, has a variance given by $\mathbb{V}\left(\hat{\mu}_n^{SS}\right) = \sum_{i=1}^{k} \frac{\lambda_i^2 \sigma_i^2}{n_i}$. Assuming the variances are known *a priori* we can minimize the error by selecting the sample sizes for each strata so that they are each approximated uniformly well. In particular, variance is minimized with the parameterization $n_i^* \propto \lambda_i \sigma_i$ which gives

$$\mathbb{V}\left(\hat{\mu}_n^{SS^*}\right) := \frac{\left(\sum_{i=1}^{k} \lambda_i \sigma_i\right)^2}{n}.$$

If all the variances are equal the above estimator has the same variance as the vanilla Monte Carlo estimate however if the variances across the strata are unequal the estimator can considerably more efficient. One practical challenge, however, is that an obvious partitioning (stratification) of the sample space is not always apparent, or the practitioner may not have a good understanding of the variability within each strata, which would permit effective sample allocations. Though, in some cases it may be worthwhile to tune these paramaters automatically while sampling; in Section 2.5 we review *adaptive stratified sampling* approaches which do just that.

2.2 Adaptive Importance Sampling

In this section we review ways in which an importance sampling proposal can be configured automatically from sampled data, as opposed to being fixed by the practitioner ahead of time.

In settings where the target distribution is unimodal and twice differentiable, a straightforward approach is to *fit* the proposal density using *moment matching* around the global

[1] Here "time" refers to the cumulative number of samples drawn for all strata.

maximum of the target density: $x^* := \arg\max_x \log(\pi(x))$. Specifically, by defining the proposal density as a convenient parametric form, such as a multivariate normal or Student density, it can be fit to a single point by setting the mean equal to x^* and covariance equal to the inverse of the Hessian matrix of $\log(\pi(x))$ at x^* (alternatively known as the *observed Fisher information matrix* I). In the case where π is well represented by the chosen parametric form this method produces, very quickly, a near optimal importance sampling distribution. Though this approach is obviously applicable to restricted class of integration problems, for some "big data" Bayesian inference tasks, this approach can be motivated by the *Bayesian central limit theorem*. This theorem essentially states that the posterior distribution converges to $\mathcal{N}(x^*, I)$ as the number of observations increases (see Sec. 4, Ghosh et al. (2006)). Coupled with data efficient optimization routines, such as stochastic gradient descent, this approach may come in useful in a number of contexts.

For more complex distributions, particularly those with multiple local modes, extending this approach raises the question of how much computational effort should be spent searching out various modes versus actual sampling. A common way to address this tradeoff is to continually re-optimize the proposal density after drawing a new sample, as is done in the *parametric adaptive importance sampling* (PAIS) scheme of Oh and Berger (1993). In particular, this PAIS approach uses a mixture of k multivariate Student densities with fixed degree of freedom ν and parameters $\lambda = \{(c_i, \mu_i, \Sigma_i)\}_{i=1}^k$ for a proposal density, that is,

$$\pi_\lambda(x) := \sum_{i=1}^{k} c_i t_\nu(x; \mu_i, \Sigma_i),$$

where $c_i \geq 0$ and $\sum_i c_i = 1$. The objective is to determine the parameterization (λ) that minimizes the variance of the importance weights $\mathbb{V}_{\pi_\lambda}(w(X))$, or equivalently $\mathbb{E}_{\pi_\lambda}\left[w(X)^2\right]$, on past data. This is done in incremental fashion where at each time-step t the algorithm draws a new sample, $X_t \sim \pi_{\lambda_t}$, then solves the optimization

$$\lambda_{t+1} = \arg\min_{\lambda} \sum_{j=0}^{t} \left(\frac{\pi(X_j)}{\pi_\lambda(X_j)}\right)^2 \frac{\pi_\lambda(X_j)}{\pi_{\lambda_j}(X_j)}. \tag{2.4}$$

The final approximation, then, is given by (2.1) or (2.2) with $w(x) = \pi(x)/\pi_{\lambda_t}(x)$. This objective function, however, can make it difficult to formulate efficient optimization routines (even in the case $k = 1$). The *efficient importance sampling* (EIS) approach of Richard and Zhang (2007) addresses this concern through the use of an alternative heuristic, specifically, the pseudo-divergence

$$d(\pi, \pi_\lambda; \alpha, \lambda) := \int (\log(\pi(x)) - \alpha - \log(\pi_\lambda(x)))^2 \pi(x) dx.$$

As with the previous approach, d can be approximated with an IS estimate using previous samples. This results in the optimization

$$\lambda_{t+1} = \arg\min_{\lambda} \min_{\alpha} \sum_{j=0}^{t} (\log(\pi(X_j)) - \alpha - \log(\pi_\lambda(X_j)))^2 \frac{\pi(X_j)}{\pi_{\lambda_j}(X_j)}.$$

This objective can be significantly easier to work with for some parameterizations of π_λ. In particular, if π_λ is a $k = 1$ mixture of exponential family distributions, the optimization reduces to a least squares problem. In this way, the EIS approach is similar to the method of *variational Bayes*: a deterministic approximation approach where one attempts to minimize the Kullback-Leibler divergence between the target and a parametric density (see Bishop et al. (2006)).

An important limitation inherent to PAIS approaches is that the target density is rarely well approximated by a single exponential family distribution. As a result, the use a mixture distribution (setting $k > 1$) is common. However, in this setting fitting the proposal is similar to solving the (NP-hard) k-means clustering problem *at each time-step*. Even if this optimization could be solved efficiently, it is not always obvious *a priori* what setting of k, or what underlying parametric distributions, might lead to a suitable approximation of the integrand.

2.2.1 Population Monte Carlo

The *population Monte Carlo* (PMC) algorithm (Cappe et al., 2004) is a clever PAIS variant that partially addresses both the issues of solving a non-convex optimization and specifying the number of modes (k) manually. These problems are side-stepped through the use of kernel density estimation on the target distribution, as opposed to a directly optimizing a parametric form. Specifically, a fixed Markov transition kernel $k_\delta(x, x')$ (defined below) parameterized by a *bandwidth* parameter δ is defined.

Definition 2.1 (Markov kernel)**.** *A Markov kernel k on the probability triple $(\mathcal{X}, \mathcal{P}, \mathcal{B})$ is a function $K : \mathcal{X} \times \mathcal{B} \to \mathbb{R}$ having the following properties.*

i. for each fixed $A \in \mathcal{B}$ the function $x \mapsto k(x, A)$ is Borel measurable.

ii. for each fixed $x \in \mathcal{X}$ the function $A \mapsto k(x, A)$ is a probability measure.

Additionally as is common in the Monte Carlo literature, for any $x, x' \in \mathcal{X}$ we let $k(x, x')$ denotes the conditional density of the transition from x to x'; that is, for any $A \in \mathcal{B}$ and we have $\mathbb{P}(X \in A|x) = \int_A k(x, x')dx'$.

The PMC approach proceeds as follows: given an initial *population* of samples $(X_1^{(0)}, ..., X_n^{(0)}) \overset{\text{iid}}{\sim} \pi_0(\cdot)$, the PMC proposal density at time-step $t+1$ is parameterized as a mixture density:

$$\pi_{t+1}(x) := \frac{1}{\mathcal{Z}} \left(\beta \sum_{i=1}^{n} w_t(X_i^{(t)}) k_\delta(x, X_i^{(t)}) + (1-\beta) t_\nu(x; \lambda, \Sigma) \right). \tag{2.5}$$

Where, $w_t(x) = \frac{\pi(x)}{\pi_t(x)}$, t_ν is a defensive sampling Student distribution with fixed parameters (ν, λ, Σ) and mixing coefficient $\beta \in [0, 1]$, and $\mathcal{Z}$ is a known normalizing constant. After executing the PMC procedure is outlined in Algorithm 1 and collecting the set of samples $\{X_{1:n}^{(0:m-1)}\}$, one may use the PMC estimator given by

$$\hat{\mu}_{PMC}^{n,m} := \frac{1}{nm} \sum_{t=0}^{m-1} \sum_{i=1}^{n} w_t(X_i^{(t)}) h(X_i^{(t)}).$$

Algorithm 1 Population Monte Carlo (PMC)

1: **for** $t \in \{0, ..., m-1\}$
2: **for** $i \in \{1, ..., n\}$
3: Sample $X_i^{(t)} \sim \pi_t(\cdot)$;
4: Compute $w_t(X_i^{(t)}) = \frac{\pi(X_i^{(t)})}{\pi_t(X_i^{(t)})}$;
5: **end for**
6: **end for**

Despite the fact that the samples drawn at each successive iteration are correlated, the PMC estimator can be shown to be unbiased through repeated applications of the tower rule. Critically, however, this statistical dependence prevents one from achieving consistency as $m \to \infty$. Instead in order to provide any theoretical guarantees one must rely on the fact that this estimate converges at a $O(n^{-1})$ rate. That is, the size of the population – which must be held in memory – must tend toward infinity to achieve convergence, which obviously presents practical challenges.

The PMC algorithm may also employ a resampling procedure where at each time-step t the population, $(X_0^{(t)}, ..., X_n^{(t)})$, is *resampled* in proportion to the weights, $(w_t(X_0^{(t)}), ..., w_t(X_n^{(t)}))$. Specifically, given a set of resampled points, denoted as $(\bar{X}_0^{(t)}, ..., \bar{X}_n^{(t)})$, the proposal is defined as

$$\pi_{t+1}(x) := \frac{1}{\mathcal{Z}} \left(\beta \sum_{i=1}^{N} k_\delta(x, \bar{X}_i^{(t)}) + (1-\beta) t_\nu(x; \lambda, \Sigma) \right). \tag{2.6}$$

In this light, the PMC algorithm is very similar to *sequential Monte Carlo* (SMC) approaches (see Doucet et al. (2001)), which often benefit tremendously from resampling.

However similar, these methods differ in one critical aspect: the importance weight in a SMC implementation at time t is given as a product of previous importance weights, $w_t(X_i^{(t)}) = \prod_{k=0}^{t} \frac{\pi(X_i^{(k)})}{\pi_k(X_i^{(k)})}$);[2] while the weights in the PMC sampler do not involve products. Consequently, the resampling procedure offers far less practical advantage in the PMC setting than the SMC setting. In fact, one can observe that the sampling from the proposal (2.5) is procedurally identical to sampling from (2.6), that is to say, the resampling step is already implicit.

The sequential nature of the PMC algorithm permits an additional form of adaptation where the kernel bandwidth δ can be tuned alongside the proposal. This is the approach taken by the *d-kernel population Monte Carlo* algorithm (Douc et al., 2007a). This adaptation is achieved by favouring the kernels, out of a discrete set, which have historically lead to higher importance weights. Given a set of kernels, $(k_1, ..., k_l)$, (having different bandwidth parameters, for example), we redefine the kernel density function used in (2.5) as a mixture of densities

$$k^{(t)}(x, x') := \sum_{j=1}^{l} \frac{\alpha_j^{(t)}}{\sum_i \alpha_i^{(t)}} k_j(x, x').$$

Where the mixture coefficients $(\alpha_1^{(t)}, ..., \alpha_l^{(t)})$ are initialized uniformly and subsequently set proportional to the sum of all past sampling weights, that is:

$$\alpha_j^{(t+1)} := \alpha_j^{(t)} + \sum_{i=1}^{n} w_t(X_i^{(t)}) \mathbb{I}\{k_i^{(t)} = k_j\}.$$

Where the indicator $\mathbb{I}\{k_i^{(t)} = k_j\}$ evaluates to 1 if density k_j is used to generate sample $X_i^{(t)}$. While the d-kernel PMC method considers only a discrete set of kernels, these ideas can be combined with the optimization routines used in PAIS. This is done in order to adapt both the mixture coefficients as well as the parameters for each kernel (bandwidth), as is done in the PMC-E algoritm of Cappé et al. (2008). Both these adaptive approaches can be shown to asymptotically converge to a proposal density (as $n \to \infty$) within the parametric class that minimizes the KL-divergence with the target distribution.

It is important to note that when choosing the value for m the practitioner must be cautious of the fact that the accuracy of the kernel density estimate will begin to diminish as the number of time steps increases, regardless of whether resampling is used. This degradation is different, but not entirely unlike, the *particle degeneracy* problem often observed in *sequential Monte Carlo* (SMC) settings. Consequently, the authors recommend setting this

[2] We review the SMC framework in Section 2.4

parameter to a rather small value, somewhere between 5-10, although this does somewhat limit the adaptability of the algorithm.

In either form, the PMC algorithm is an effective and elegant way of fitting the proposal density without requiring a non-convex optimization. That said, in practice, PMC improves on standard PAIS only to the extent that kernel density estimation is easier than the k-means clustering. There are certainly domains where this is the case, but in general both of these tasks are computationally intractable and, consequently, both methods suffer from the various effects of local minima.

2.2.2 Discussion

An interesting aspect of the adaptive importance sampling setting is the fact that jointly optimizing the proposal, and integrating the function, immediately gives rise to an *exploration-exploitation tradeoff*. In particular, if a given algorithm converges too quickly on any mode of π, it will no longer sample other parts of the domain. It will therefore not be able to correct for the improper fit. This, coupled with the fact that very-high variance IS samplers are difficult to detect numerically, can result in highly unstable behaviour. The standard approach to address this is to mix the proposal with a defensive sampling distribution, though this heavy-handed approach ultimately results in unnecessary exploration and prevents asymptotic convergence to the optimal proposal. Developing methods for addressing this tradeoff remains an active area of research and the methods presented in Chapter 4 can be seen, in part, as early steps in this area.

Perhaps a more fundamental observation to be made here is that adaptive importance sampling strategies can be highly advantageous for simple integration tasks (i.e. small number of modes/dimensions) but often fail dramatically when presented with more complex problems. The unfortunate reality is that the core ideas of such *global fitting* approaches are somewhat at odds with the initial motivations for Monte Carlo integration. That is, Monte Carlo methods are often deployed as a last resort in complex domains where alternative approaches do not perform well. It is no coincidence that these problems are, overwhelmingly, those where the intergrand is not easily approximated with convenient parametric, or semi-parametric, forms.

The main alternative to these global fitting approaches are what we refer to as *local move methods* which exploit local structure through the use of simulated Markov chains. At a high level this strategy forms a basis for both the popular Markov chain Monte Carlo approach and the sequential Monte Carlo approach, which we review in the following sections.

2.3 Markov Chain Monte Carlo

The Markov chain Monte Carlo (MCMC) approach is possibly the most recognized and widely used Monte Carlo approach. As a result of its fundamentally different sampling mechanics, the method has its own rich body of literature which includes a number of unique strategies having a wide variety of applications. Most of this work falls outside our scope and in this section we will only touch on some key aspects of MCMC as they relate to the methods we examine in this thesis. For a broader survey on the MCMC approach we refer to the seminal review given by Neal (1993) as well as the textbook treatments of Liu (2001) and Robert and Casella (2005). Additionally, we note that some of the more cutting edge developments in MCMC include methods for exploiting gradient information on the target density, such as Langevin-adjusted and Hamiltonian MCMC as summarized by Neal (2011). Additionally, notable extensions to this work include methods for exploiting Riemann geometry (Girolami and Calderhead, 2011) and *mini-batch* methods for "big data" applications (Ahn et al., 2012; Bardenet et al., 2014). Additionally, *adaptive* MCMC methods, where the parameters of the simulation are tuned as a function of past samples, are now fairly well understood and have lead to a number of practical advancements (Andrieu et al., 2006; Roberts and Rosenthal, 2009).

In general, MCMC methods can be used to generate samples from any distribution, even if known only up to a normalizing constant, through the simulation of a Markov chain having a stationary distribution equal to this target. Much of the popularity of this approach can be attributed to the ease at which such a Markov chain may be constructed. In particular, one must ensure that the chain satisfies condition of *detailed balance*:

$$T(x, x')\pi(x) = T(x', x)\pi(x'),$$

were T is a Markov kernel that defines the simulated chain. Additionally, it must be ensured that the chain is *ergodic*, which implies that the chain will always result in the same stationary distribution regardless of starting point; in a way, this condition is analogous to the support conditions in importance sampling. If these two conditions are met, then samples generated from this simulation are guaranteed to be distributed (asymptotically) according to π.

The most basic MCMC construction is the *Metropolis-Hastings* (MH) sampler which uses the transition kernel $T(x, x') = k(x, x')\alpha(x, x')$ where $k(x, x')$ is a fixed Markov kernel (the *proposal*) and $\alpha(x, x')$ is an acceptance function which gives the probability of

accepting or *rejecting* a given move so as to satisfy detailed balance, that is

$$\alpha(x, x') := \min\left(1, \frac{\pi(x')k(x', x)}{\pi(x)k(x, x')}\right).$$

The algorithm proceeds as follows, given the current point X_t, we propose point $X'_t \sim k(X_t, \cdot)$, and sample $Y_t \sim \text{Uniform}(0, 1)$ we then let

$$X_{t+1} = \mathbb{I}\{Y_t < \alpha(X_t, X'_t)\}X'_t + \mathbb{I}\{Y_t \geq \alpha(X_t, X'_t)\}X_t\,.$$

Note that the proposal k is unlike an importance sampling proposal in that it does not have full support over π and instead, as with PMC, has a common starting point that is a Gaussian kernel of specified width. Approximations of (1.1) may then be computed by simulating the above Markov chain, starting from an arbitrary start point X_0, for some specified number of time-steps, n, and evaluating the empirical sum

$$\hat{\mu}^n_{MCMC} := \frac{1}{n - n_B}\sum_{i=n_B}^{n} h(X_i). \tag{2.7}$$

Here n_B is the number of samples required to surpass the *burn-in* period and should be large enough to ensure the X_{n_B} is independent of X_0. As one can imagine, in settings where the target density is concentrated in a small region of the sample space, or along a lower-dimensional manifold, the MH algorithm will conduct a random walk over the relevant areas while largely ignoring the irrelevant parts. In high dimensional problems, this integrand structure is common and, as a result, the performance of MCMC is simply unrivalled for such tasks.

It is possible to derive convergence guarantees for MCMC approaches generally through conditions on the *mixing rate* of the Markov chain. That is, the rate at which the dependency between samples X_i and X_{i+d} drops off as d increases. For instance, if two samples were guaranteed to be independent after some finite number of steps (d) it is clear that the MSE of (2.7) decreases at a rate of $O(d/n)$. In practice, however, it can be challenging to engineer rapidly mixing Markov chains. Particularly in cases where the target density is comprised of multiple modes separated by regions of low probability. Also, poor mixing is often difficult to diagnose numerically as there is no good way of knowing whether all of the modes have been visited, and in the correct proportions.

Poor mixing rates may be one reason to favour an importance sampling approach over MCMC. Additionally, importance sampling approaches are often favoured for approximating the normalization constant for $\hat{\pi}$, as this task can be quite challenging with MCMC approaches. For instance, the most straightforward strategy, which is likely the first thing any

unsuspecting practitioner might try, is the so-called *harmonic mean* (HM) method (Newton and Raftery, 1994). Here, one first collects samples $X_i \sim \pi$ using any MCMC approach, then uses these samples in conjunction with the weighted importance sampling estimator to approximate the normalizing constant. Specifically, one deploys the estimator:

$$\hat{\zeta}_{HM}^{n} := \left(\frac{1}{n - n_B} \sum_{i=n_B}^{n} \frac{1}{\hat{\pi}(X_i)} \right)^{-1}.$$

Although intuitive and mathematically elegant, in practice this estimate is rarely useful as it will typically exhibit extremely high or infinite variance. There are numerous other ways of computing normalizing constants with MCMC (see Neal (2005)) though none can be said to be as straightforward or generally applicable as their counterparts for approximating μ.

2.4 Sequential Monte Carlo

Another sophisticated and widely applicable class of Monte Carlo algorithms are sequential Monte Carlo (SMC) methods. In the SMC setting the distribution of interest, π, is assumed to be decomposable into a sequence of known conditional densities, that is

$$\pi(x^{(0:m)}) = \pi_0(x^{(0)})\pi_1(x^{(1)}|x^{(0)}) \dots \pi_n(x^{(m)}|x^{(m-1)}).$$

The integral in (1.1) can then be written as

$$\mu = \int h(x^{(0:m)})\pi(^{(0:m)})dx^{(0:m)} = \int h(x^{(0:m)})\pi_0(x^{(0)}) \prod_{i=1}^{m} \pi_i(x^{(i)}|x^{(i-1)})dx^{(0:m)}.$$

Additionally, it is often the case that h may factored similarly to the above equation or is independent of most variables. For instance, it is common that $h(x^{(0:m)}) = h(x^{(m)})$. The SMC formulation arises naturally in numerous practical settings from protein formation, robot dynamics, to financial option pricing (see Doucet et al. (2001)). Similar to the general integration problem, it is often the case that direct simulation from π_i does not result in efficient estimators. In such cases the sequential importance sampling (SIS) approach may be used where one uses a similarly factored proposal density: $g(x^{(0:m)}) := g_0(x^{(0)})g_1(x^{(1)}|x^{(0)}) \dots g_n(x^{(m)}|x^{(m-1)})$. Given samples $(X_1^{(1:m)}, ..., X_n^{(1:m)}) \overset{\text{iid}}{\sim} g(\cdot)$ we arrive at the following estimator

$$\hat{\mu}_n^{SIS} := \frac{1}{n} \sum_{i=1}^{n} w(X_i^{(0:m)})h(X_i^{(0:m)}),$$

where,

$$w(x^{(0:m)}) = \left(\frac{\pi_0(x^{(0)})}{g_0(x^{(0)})} \right) \left(\frac{\pi_1(x^{(1)}|x^{(0)})}{g_1(x^{(1)}|x^{(0)})} \right) \dots \left(\frac{\pi_m(x^{(m)}|x^{(m-1)})}{g_m(x^{(m)}|x^{(m-1)})} \right).$$

This approach is often desirable either because conditional proposal densities are easier to engineer or because population-based resampling approaches can be used, as is done by the *sequential importance resampling* (SIR) algorithm of Gordon et al. (1993). The insight into these populations-based techniques is as can be explained as follows. Suppose we have samples $(X_1^{(j-1)}, ..., X_n^{(j-1)}) \overset{\text{iid}}{\sim} \pi_{j-1}(\cdot|X^{0:j-2})$ as well as samples $\{X_i^{(j)} \sim g_j(\cdot|X_i^{(j-1)})\}_{i=1}^n$. Given this, one can approximate the distribution π_j with the empirical distribution

$$\pi_j(x|x^{(0:j-1)}) \approx \sum_{i=1}^n \bar{w}_j(X_i^{(j)})\delta_{X_i^{(j)}}(x),$$

where $\bar{w}_j(X_i^{(j-1)}) := \frac{w_j(X_i^{(j-1)})}{\sum_{k=1}^n w_j(X_k^{(j-1)})}$ with $w_j(X_i^{(j-1)}) = \frac{\pi_j(X_i^{(j)}|X_i^{(j-1)})}{g_j(X_i^{(j)}|X_i^{(j-1)})}$ and δ_x denotes the Dirac δ-function centered at x. This empirical distribution may used as-is to simulate the next step of the recursion. However, over time, the variance of the importance weights will tend to increase which results in a poor approximation. This problem is referred to as *particle degeneracy* and is routinely measured by the *effective sample size* (ESS) (Liu and Chen, 1998) given by

$$\text{ESS}(X_{1:n}) := \frac{\left(\sum_{i=1}^n w(X_i)\right)^2}{\sum_{i=1}^n w(X_i)^2}. \tag{2.8}$$

The ESS takes on values in $[1, n]$ and can be used to monitor the health of the population. If the value drops below some fixed threshold, $\varepsilon \in (0, n)$, one may attempt improve the approximation by *resampling* a new set of particles from the empirical distribution. The most straightforward resampling approach is to draw particles with replacement from a multinomial distribution, though there are a number of more efficient resampling procedures such as residual (Liu and Chen, 1998), systematic (Carpenter et al., 1999), and stratified resampling (Kitagawa, 1996). A basic SIR implementation, which uses simple multinomial resampling at every step, is given in Algorithm 2.

Following normal execution of Algorithm 2 the SIR estimator for μ is given as

$$\hat{\mu}_n^{SIR} := \frac{1}{n}\sum_{i=1}^n h(X_i^{(m)})w_m(X_i^{(m)}),$$

as well as the normalizing constant (see Del Moral and Doucet (2002)),

$$\hat{\zeta}_n^{SIR} := \sum_{j=0}^m \log\left(\sum_{i=1}^n w_j(X_i^{(j)})\right) - m\log n.$$

Equivalent estimators for the case where resampling stages are executed depending on the effective sample size are given in Del Moral et al. (2006).

Algorithm 2 Sequential Importance Resampling (SIR)

for $i \in \{1, ..., n\}$
  Sample $X_i^{(0)} \sim g_0(\cdot)$;
  Compute $w_i^{(0)} = \frac{\pi_0(X_i^{(0)})}{g_0(X_i^{(0)})}$;
end for
for $j \in \{1, ..., m\}$
  Resample $\bar{X}_{1:n}^{(j-1)} \sim \text{Multinomial}(X_{1:n}^{(j-1)}, \bar{w}(X_{1:n}^{(j-1)}), n)$;
  for $i \in \{1, ..., n\}$
    Sample $X_i^{(j)} \sim g_j(\cdot|\bar{X}_i^{(j-1)})$;
    Compute $w_j(X_i^{(j)}) = \frac{\pi_j(X_i^{(j)}|\bar{X}_i^{(j-1)})}{g_j(X_i^{(j)}|\bar{X}_i^{(j-1)})}$;
  end for
end for

In general, the use of resampling in SMC methods can lead to considerably improved estimates for some problem domains. In particular, sequential state tracking tasks. However, the approach is not guaranteed to offer a reduction in variance and may actually introduce additionally approximation errors due to resampling variance and particle degeneracy. Additionally, as with the population Monte Carlo approach theoretical convergence results, typically in the form of a central limit theorem, require that the population size grow indefinitely.

2.4.1 Sequential Monte Carlo Samplers

Sequential Monte Carlo samplers (SMCS) (Del Moral et al., 2006) are a class of importance sampling algorithms that define their proposal distributions as a sequence of local Markov transitions. As a result of this construction, SMCS methods are able to combine many of the advantages of MCMC methods with those of SMC methods. SMCS algorithms operate under the same assumptions as regular importance sampling in that they do not require that the target distribution factors into a product of conditionals. Instead, the algorithm exploits the fact that a sequence of Markov transitions may be factored in much the same way.

Specifically, the SMCS proposal density is defined using an initial proposal, π_0, as well as a sequence of *forward* Markov kernels, $f_{1:m}$, which are assumed to be evaluable pointwise and simulable. In particular, we define the proposal density as

$$g(x^{(0:m)}) := \pi_0(x^{(0)}) \prod_{j=1}^{m} f_j(x^{(j-1)}, x^{(j)}). \tag{2.9}$$

In order to permit convenient cancellations in later steps, the target distribution, π, is also augmented with a sequence of *backward* Markov transition kernels, $b_{1:m}$, which are re-

quired only to be efficiently evaluable point-wise, that is, we define

$$\widetilde{\pi}(x^{(0:m)}) := \pi(x^{(m)}) \prod_{j=1}^{m} b_j(x^{(j)}, x^{(j-1)}). \tag{2.10}$$

Since each $b_j(x, \cdot)$ is a probability distribution we have $\int b_j(x, x')dx' = 1$ it then follows that $\int \widetilde{\pi}(x^{(0:m)})dx^{(0:m-1)} = \pi(x^{(m)})$, which implies that $\int h(x^{(m)})\widetilde{\pi}(x^{(0:m)})dx^{(0:m)} = \mu$. From here, we can approximate this expanded integral through importance sampling methods, which can be verified by observing

$$\begin{aligned}\int \frac{h(x^{(m)})\widetilde{\pi}(x^{(0:m)})}{g(x^{(0:m)})} g(x^{(0:m)})dx^{(0:m)} &= \int h(x^{(m)})\widetilde{\pi}(x^{(0:m)})dx^{(0:m)} \\ &= \int h(x^{(m)})\pi(x^{(m)})dx^{(m)}.\end{aligned}$$

A critical detail is that $\widetilde{\pi}$ and g are easily factored, which will later permit the application of SMC methods (i.e. resampling).

In order to build in additional flexibility, the formulation allows for the use of local moves in conjunction with annealing distributions. Specifically, a sequence of unnormalized *auxiliary distributions*, $\pi_{0:m}$, which are generally expected to blend smoothly between a the initial proposal distribution (π_0) and the target distribution ($\pi_m := \pi$) are used. A common choice is the *tempered* version $\pi_j = \pi^{(1-\beta_j)}\pi_0^{\beta_j}$ for some fixed annealing schedule $1 = \beta_0 > \beta_1 > ... > \beta_m = 0$ (Neal, 2001; Gelman and Meng, 1997). Additionally, in Bayesian settings the sequence of posterior distributions each with incrementally more data may be useful, that is, $\pi_i(x) = \pi(x|z_1, ..., z_{(1-\beta_i)l})\pi_0(x)$ where $(z_1, ...z_l)$ denotes the set of observed data (Chopin, 2002). Of course, the use of annealed distributions is not strictly required and one may elect to use the *homogenous* parameterization, where $\pi_1 = ... = \pi_n = \pi$. Using these auxiliary distributions the target distribution is then re-written as

$$\hat{\pi}(x^{(0:m)}) := \prod_{j=1}^{m} \frac{\pi_j(x^{(j)})b_j(x^{(j)}, x^{(j-1)})}{\pi_{j-1}(x^{(j-1)})},$$

where one can observe that these auxiliary distributions telescope to give $\hat{\pi}(x^{(0:m)}) = \frac{\widetilde{\pi}(x^{(0:m)})}{\pi_0(x^{(0)})}$. Temporarily ignoring these potential cancellations, the stepwise importance weighting function can be defined as

$$r_j(x^{(j-1)}, x^{(j)}) := \frac{\pi_j(x^{(j)})b_j(x^{(j)}, x^{(j-1)})}{\pi_{j-1}(x^{(j-1)})f_j(x^{(j-1)}, x^{(j)})}. \tag{2.11}$$

Using these formula the basic SMC procedure for simulating from g and recursively computing the appropriate weighting is given in Algorithm 3. As with standard SMC, it is straightforward to add in a resampling step to this procedure.

Algorithm 3 Sequential Monte Carlo Sampling (SMCS)

for $i \in \{1, ..., n\}$
  Sample $X_i^{(0)} \sim \pi_0(\cdot)$;
  Compute $W_i^{(0)} = \frac{\pi_1(X_i^{(0)})}{\pi_0(X_i^{(0)})}$;
end for
for $j \in \{1, ..., m\}$
  for $i \in \{1, ..., n\}$
    Sample $X_i^{(j)} \sim f_j(X_i^{(j-1)}, \cdot)$;
    Compute $W_i^{(j)} = W_i^{(j-1)} r_j(X_i^{(j-1)}, X_i^{(j)})$
  end for
end for

After generating samples $\{X_{1:n}^{(0:m)}\}$ and corresponding weights $\{W_{1:n}^{(0:m)}\}$ with this procedure the (unbiased) SMCS estimator is given by

$$\hat{\mu}_n^{SMCS} := \frac{1}{n} \sum_{i=1}^{n} W_i^{(m)} h(X_i^{(m)}),$$

additionally, as with standard importance sampling, the normalization constant for π may be estimated as

$$\zeta_n^{SMCS} := \frac{1}{n} \sum_{i=1}^{n} W_i^{(m)}.$$

In the case where *homogenous* auxiliary distributions are used the samples at each time-step may be used (Del Moral and Doucet, 2002), resulting the following (unbiased) estimator

$$\hat{\mu}_n^{SMCS'} := \frac{1}{n(m+1)} \sum_{i=1}^{n} \sum_{j=0}^{m} W_i^{(j)} h(X_i^{(j)}),$$

with $\zeta_n^{SMCS'}$ defined similarly.

The most widely deployed instantiation of SMCS is the pre-dated method of *annealed importance sampling* (Neal, 2001), which can recovered by parameterizing the backward kernel as

$$b_j(x, x') = \frac{\pi_j(x')}{\pi_j(x)} f_j(x', x),$$

where f_j is any valid MCMC transition for π_j; that is, $f_j(x, x')\pi_j(x) = f_j(x', x)\pi_j(x')$. This choice leads to cancellations in the importance weighting function, yielding the simpler weighting function $r_j(x, x') = r_j(x) = \frac{\pi_j(x)}{\pi_{j-1}(x)}$.[3]

This particular parameterization is powerful because it opens the door for the application of vast number of existing MCMC strategies. Moreover, AIS is able to sidestep the

[3] Multiple MCMC transitions are typically executed on the same annealing "step", as the composition still satisfies detailed balance.

drawbacks of MCMC associated with mixing, since we have not required that the Markov chain simulate for some *burn in* period or approach stationarity to ensure unbiasedness. Unfortunately however, the MCMC chain must still produce samples roughly distributed according to the stationary distribution in order for the approach to offer a meaningful variance reduction (see Neal (2001)). It is worth noting that in cases where π does not permit efficient mixing AIS may be advantageous since the annealed distributions π_i may be significantly easier to simulate with MCMC. In fact, this annealing is the key intuition behind the widely deployed MCMC method of *parallel tempering* (Neal, 1996; Earl and Deem, 2005).

2.4.2 Adaptive SMCS

SMCS algorithms can be considerably more effective than simple importance sampling methods since the proposal is altered automatically by the observed values of the integrand. Despite this dependence, SMCS methods are generally not considered *adaptive* methods since the parameters of the algorithm – i.e. the MCMC moves, annealing rate, initial proposal – remain fixed. However, as with previous approaches there have been a handful of *adaptive SMCS* extensions proposed in the literature. That is, procedures for automatically tuning the parameters of a SMCS algorithm using data obtained in past simulations (Chopin, 2002; Schäfer and Chopin, 2013; Jasra et al., 2011).

As it currently stands, there is no general approach for incorporating such adaptive behaviour into SMCS; the few existing methods have little in common and each rely on specific properties of the underlying SMCS method. That said, provided that the adaptation occurs under mild technical conditions it is at least straightforward to obtain asymptotic convergence results, in the form of consistency proofs for the resulting estimators, for any adaptive SMCS approach (Beskos et al., 2013). We note also that though all of the existing approaches make use of particle representations and resampling as the primary engine driving this adaptation, this is not strictly necessary.

One popular adaptive SMCS approach is given by Schäfer and Chopin (2013) who proposed a method for sequentially tuning the proposal for a Metropolis-Hastings MCMC sampler used by SMCS. In particular, it was observed that the ideal MH proposal at time t is given by the distribution π_t. Unfortunately, this distribution cannot be sampled from efficiently, otherwise we would not be using MCMC in the first place. In order to bypass this difficulty, the proposed approach *fits* a parametric distribution to the current particle representation of π_t through straightforward optimization routines. This fit proposal is then used

in the MCMC transitions for the next update to the particle representation, and the process is repeated. Fitting a parametric density to a set of particles is a non-convex optimization in general, so some form of kernel density estimation may be advantageous.

Perhaps the most effective, and widely applicable, adaptive SMCS strategy is to tune the set of auxiliary distributions using the particle representation. The rate at which these distributions converge to the target density has a significant effect on the performance of the SMC sampler. If the distributions converge too slowly the sampler will achieve less variance reduction per unit of computation. On the other hand, if the distributions converge too quickly, poor MCMC mixing or high variance importance weights (which leads to particle degeneracy) often result. With these considerations in mind, one approach, given by Schäfer and Chopin (2013), is to tune the annealing parameters at each step so that a fixed effective sample size (Eq. (2.8)) is maintained, for example $\text{ESS}^* = 0.8n$.

In particular, supposing that *tempered* distributions are used with β_j defined as $\beta_{j+1} = \beta_j + \alpha$ for some fixed $\alpha \in (0, 1)$, we have that $w_j(x, x') = r_j(x) = \frac{\pi_j(x)}{\pi_{j-1}(x)} = \frac{\pi(x)^\alpha}{\pi_0(x)^\alpha}$. The ESS then can be expressed as a function of α and the population $X_{1:n}^{(j)}$ and we can solve for α^* s.t. $ESS(\alpha^*, X_{1:n}^{(j)}) = \text{ESS}^*$. As with the adaptive approach above, this procedure can be shown to offer improved empirical performance over fixed schedules while still providing asymptoticly consistent estimators (Beskos et al., 2013).

2.5 Adaptive Stratified Sampling

As mentioned previously, the method of stratified sampling can yield substantial reductions in variance if the stratification and sample allocations are well tuned. However, in order for the allocation of samples to be efficient the standard deviation (σ_i) of individual samples within a given strata must be known beforehand, which is rarely the case. A natural strategy then, is to approximate these standard deviations from previous samples and adjust future allocations accordingly.

One such *adaptive stratified sampling* strategy is proposed by Etoré and Jourdain (2010) and operates similar to the sequential optimization procedures for adaptive importance sampling detailed previously. In particular, this approach proceeds in a series of *stages* where the number of samples drawn in each stage (l) is specified beforehand. The first step in each stage is to approximate the sample standard deviations for each strata as (reusing the

notation from Section 2.1.3):

$$\hat{\sigma}_{i,n} := \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i} X_{i,j}^2 - \hat{\mu}_{i,n}^2}\,,$$

where $i \in \{1, ..., k\}$ denotes the strata index, n_i the number of samples allocated to strata i thus far, $n = \sum_i n_i$, $X_{i,j} \sim \pi_i(\cdot)$, and $\hat{\mu}_{j,n} := \frac{1}{n_i}\sum_{j=1}^{n_i} X_{i,j}$. The next step is the *exploration* phase where a single draw is allocated to each strata, which ensures the asymptotic convergence of $\hat{\sigma}_{i,n}$ to σ_i. The remaining $l - k$ samples are then allocated in an *exploitive* manner. That is, in proportion to the optimal value assuming the $\hat{\sigma}_{i,n}$ approximations are accurate ($n_i \propto \lambda_i \hat{\sigma}_{i,n}$). Etoré and Jourdain provide asymptotic analysis for this sampling scheme and show that each strata is sampled in the optimal proportion so long as the *exploration* phase ensures each stata is sampled infinitely often and yet takes up a negligible percentage of the samples in relation to the *exploitation* phase.

More recently, there have been a number of developments in adaptive stratified sampling in the field of machine learning which have borrowed techniques for balancing exploration and exploitation from the *multi-armed bandit* literature (Grover, 2009; Carpentier and Munos, 2011; Carpentier, 2012). Algorithmically, these approaches are not entirely unlike the approach of Etoré and Jourdain, though they balance this tradeoff more carefully. As a result, these approaches can be shown to achieve improved empirical performance as well as permit more robust finite-time guarantees.

The problem is formalized as follows: consider a sequential allocation algorithm $\mathcal{A}$ to be a procedure for selecting, at each time-step t, which strata should receive the next sample. Denote this choice as $I_t \in \{1, ..., k\}$. One can write the loss (MSE) of the stratified sampling estimator for $\mathcal{A}$ as:

$$\bar{L}_n(\mathcal{A}) := \mathbb{E}\big[(\hat{\mu}_n - \mu)^2\big] = \mathbb{E}\left[\left(\sum_{i=1}^{k} \lambda_i(\hat{\mu}_{i,n} - \mu)\right)^2\right].$$

Letting the random variable $T_{i,n} := \sum_{t=1}^{n} \mathbb{I}\{I_t = i\}$ denote the number of samples for strata i up to time n one observes that the estimate $\hat{\mu}_{i,n} := \frac{1}{T_{i,n}}\sum_{t=1}^{T_{i,n}} X_{i,t}$ is not necessarily unbiased since $T_{i,n}$ is allowed to depend on $X_{1:n-1}$. This dependence is the key frustration in analyzing these algorithms – indeed most bandit algorithms – and motivates the use of the simplified *weighted MSE loss* $L_n(\mathcal{A})$ defined as the first term in the expanded loss:

$$\bar{L}_n(\mathcal{A}) = \underbrace{\mathbb{E}\left[\sum_{i=1}^{k} \lambda_i^2(\hat{\mu}_{i,n} - \mu)^2\right]}_{L_n(\mathcal{A})} + \mathbb{E}\left[2\sum_{i\neq j}^{k} \lambda_i\lambda_j(\hat{\mu}_{i,n} - \mu)(\hat{\mu}_{j,n} - \mu)\right].$$

Additionally, since we are often interested in the performance of our allocation algorithm with respect to loss under the optimal allocation $L_n(\mathcal{A}^*) = \frac{(\sum_{i=1}^k \lambda_i \sigma_i)^2}{n}$ (Section 2.1.3), we define the *regret* for algorithm $\mathcal{A}$ as

$$R_n(\mathcal{A}) = L_n(\mathcal{A}) - L_n(\mathcal{A}^*) \quad \text{also} \quad \bar{R}_n(\mathcal{A}) = \bar{L}_n(\mathcal{A}) - L_n(\mathcal{A}^*).$$

The first algorithm in this space was the GAFS-WL algorithm proposed by Grover (2009), which allocated samples in the optimal proportion $\lambda_i \hat{\sigma}_i$ while managing exploration by ensuring each strata sampled at least $\sqrt{t}$ times. This simple algorithm was shown to have regret bounded as $R_n(\mathcal{A}) = \widetilde{O}(n^{-3/2})$ for any n, where the notation $\widetilde{O}$ hides logarithmic factors. An alternative approach to managing exploration and exploitation in bandit problems is to construct confidence bounds around the approximation $\hat{\sigma}_i$ and allocate samples according the worst-case (highest variance) value. Using this strategy one arrives at the MC-UCB algorithm of Carpentier and Munos (2011), which can also be shown have a regret bounded as $R_n(\mathcal{A}) = \widetilde{O}(n^{-3/2})$ though with slightly different constant factors than GAFS-WL. This result was later improved upon and it can be shown that the "true" regret is bounded as $\bar{R}_n(\mathcal{A}) = \text{poly}(\beta_{min}^{-1})\widetilde{O}(n^{-3/2})$, where β_{min}^{-1} is a problem-dependent constant. Additionally, for the problem-independent case (minimax) the regret can be bounded as $\bar{R}_n(\mathcal{A}) = \widetilde{O}(n^{-4/3})$ which has a matching lower bound (up to log factors), that is, $\bar{R}_n(\mathcal{A}) = \Omega(n^{-4/3})$ for *any* algorithm (Carpentier et al., 2014).

The strong finite-time theoretical guarantees paired with these algorithms is in many ways the main contribution of this work since this form of analysis is rarely seen in the area of adaptive Monte Carlo.

2.6 Summary

In this chapter we have reviewed a number of the more powerful Monte Carlo approaches and variance reduction techniques as well as adaptive extensions to many of these approaches. In the following chapters we present novel approaches extending many of the methods presented here.

Chapter 3

Variance Reduction via Antithetic Markov Chains

"There are no routine statistical questions, only questionable statistical routines."

– D.R. Cox

In this chapter we introduce a novel approach, dubbed *antithetic Markov chain sampling* (AMCS) (Neufeld et al., 2015), which improves on a fixed proposal density through the addition of local Markov chain simulations. In this respect, the approach is similar to the sequential Monte Carlo sampling approach of Del Moral et al. (2006) and, at a high level, both approaches are essentially exploiting the same observation: integrands are often relatively smooth and have the majority of their mass concentrated into local regions or modes. The approaches differ primarily in how they parameterize the Markov chains which, as we will see, ultimately affects the types of integrands for which either method is most suitable.

Specifically, from each sampled point the AMCS sampler simulates two separate, short-lived, Markov chains that are designed to head in opposite directions. The objective of these chains is to quickly acquire a set of points for which the value of the integrand takes on a large range of values. Ultimately, by averaging over these points much of the local variability in the integrand can be removed, lowering the variance of the final estimate. This technique allows for substantial savings over comparable Monte Carlo methods on highly *peaked* and multimodal integrands. An example of such an integrand – the log likelihood function for a Bayesian robot localization task which we consider in Section 3.5 – in shown in Fig. 3.1. This likelihood function has thousands of local modes separated by regions of low probability which makes it challenging to engineer efficient proposal densities as well as MCMC chains that can explore this space efficiently. However, there is still a great deal of local structure, in the form of local modes, which can be exploited by an AMCS sampler.

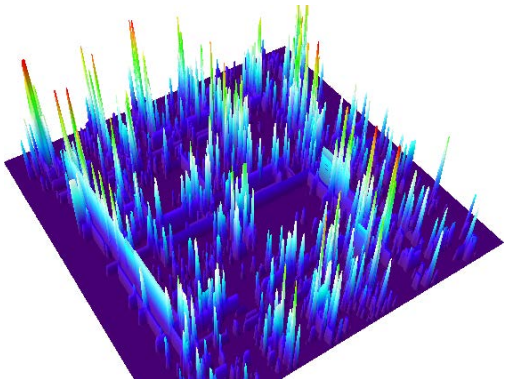

Figure 3.1: Log-likelihood function of position given sensor readings in a Bayesian robot localization problem, 2 of 3 dimensions shown.

In the remainder of the chapter we outline the mechanics of the AMCS approach and provide some specific parameterizations and motivations for the Markov chains used by the method. Additionally, we establish the unbiasedness of this approach and analyze the potential for variance reduction. Lastly, the effectiveness of the approach, in contrast to similar algorithms, is evaluated empirically on non-trivial machine learning tasks.

3.1 Approach

In an nutshell, a single iteration of the AMCS algorithm proceeds as follows: the algorithm draws a single sample from the proposal π_0, simulates two independent Markov chains to produce a set of points, evaluates the target function on each, then returns the resulting average. After executing n (independent) runs of this procedure the empirical average of all these returns is then used as the final estimate.

In what follows we refer to the two Markov chains as the *positive* and *negative* chain and denote the Markov transition kernels (Definition 2.1) used by each as k^+ and k^- respectively. Additionally, these chains terminate according to corresponding probabilistic stopping rules referred to as (positive and negative) *acceptance functions*, α^+ and α^-, which specify the probability of accepting a move in the respective directions. The kernels must be efficiently evaluable, simulable, and must also satisfy a joint symmetry property together with the acceptance functions as described in Definition 3.1. Additionally, for simplicity throughout this chapter we assume that all functions ($\alpha^{+/-}$, h) and densities are measurable as needed.

Definition 3.1. *The Markov kernels and acceptance functions specified by* (k^+, α^+) *and*

(k^-, α^-) *are said to be* jointly symmetric *iff for any* $x, x' \in \mathbb{R}^d$ *the following holds*

$$k^+(x, x')\alpha^+(x, x') = k^-(x', x)\alpha^-(x', x).$$

Using these components the AMCS procedure can be described as follows. For each of the n independent runs of this algorithm, a *starting point* $X^{(0)} \sim \pi_0(\cdot)$ is first sampled from the given proposal density. Outward from this starting point both a *positive* and *negative* Markov chain are then simulated. For the positive chain, points are generated as $X^{(j)} \sim k^+(X^{(j-1)}, \cdot)$ where $j \geq 1$, and this chain is terminated at step j with probability $\alpha^+(X^{(j-1)}, X^{(j)})$. The stopping time for this chain can be written using the acceptance variable $A^{(j)} \in \{0, 1\}$ where $A^{(j)} \sim \alpha^+(X^{(j-1)}, X^{(j)})$[1] which gives $M := 1 + \sum_{j \geq 1} \prod_{l=1}^{j} A^{(l)}$. The negative chain is denoted using indicies $j \leq -1$, where similarly define $X^{(j)} \sim k^-(X^{(j+1)}, \cdot)$, $A^{(j)} \sim \alpha^-(X^{(j+1)}, X^{(j)})$, and stopping time $N := 1 + \sum_{j \leq -1} \prod_{l=-1}^{j} A^{(l)}$. The full algorithm is outlined in Algorithm 4 and the graphical model representing the dependencies between these variables is shown in Fig. 3.2 below.

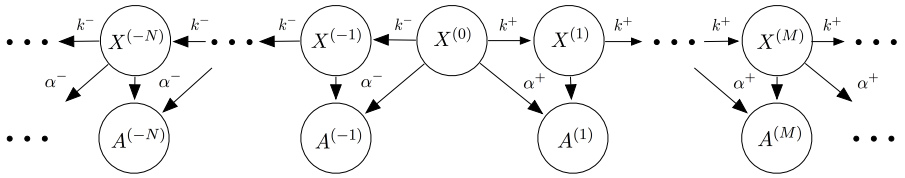

Figure 3.2: Graphical model outlining the dependencies between the sampled variables, here the positive chain $(X^{(1)}, ..., X^{(M)})$ is shown on the right of $X^{(0)}$ while the negative chain $(X^{(-1)}, ..., X^{(-N)})$ is shown on the left. Any variables corresponding to indices greater than M or less than $-N$ are not sampled by the algorithm.

After n independent trajectories have been collected, $\left\{\left(X_i^{(-N_i)}, ..., X_i^{(M_i)}\right)\right\}_{1 \leq i \leq n}$ μ and ζ may be approximated with the estimators

$$\hat{\mu}^n_{AMCS} := \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{j=1-N_i}^{M_i - 1} h(X_i^{(j)}) \pi(X_i^{(j)})}{(M_i + N_i - 1)\pi_0(X_i^{(0)})}, \tag{3.1}$$

$$\hat{\zeta}^n_{AMCS} := \frac{1}{n} \sum_{i=1}^{n} \frac{\sum_{j=1-N_i}^{M_i - 1} \hat{\pi}(X_i^{(j)})}{(M_i + N_i - 1)\pi_0(X_i^{(0)})}. \tag{3.2}$$

Note that the two endpoints $X^{(M)}$ and $X^{(-N)}$ are not used in these estimators, we refer to all the *other* points in a trajectory $(X^{(1-N)}, ..., X^{(M-1)})$ as the *accepted* points. For

[1]Here $A^{(j)} \sim \alpha^+(X^{(j-1)}, X^{(j)})$ implies $\mathbb{P}\Big(A^{(j)} = 1\Big) = \alpha^+(X^{(j-1)}, X^{(j)})$.

Algorithm 4 AMCS Procedure

for $i \in \{1, ..., n\}$
  Sample $X_i^{(0)} \sim \pi_0(\cdot)$;
  for $j = 1, 2, ...$
    Sample $X_i^{(j)} \sim k^+(X_i^{(j-1)}, \cdot)$;
    Sample $A_i^{(j)} \sim \alpha^+(X_i^{(j-1)}, X_i^{(j)})$
    If $(A_i^{(j)} = 0)$ **break loop** and set $M_i = j$;
  end for
  for $j = -1, -2, ...$
    Sample $X_i^{(j)} \sim k^-(X_i^{(j+1)}, \cdot)$;
    Sample $A_i^{(j)} \sim \alpha^-(X_i^{(j+1)}, X_i^{(j)})$
    If $(A_i^{(j)} = 0)$ **break loop** and set $N_i = -j$;
  end for
end for
return estimate from Eq. (3.1) or Eq. (3.2).

instance, in the case that the first move in both directions is rejected there is only one accepted point: $X^{(0)}$.

This generic formulation is not particularly useful without some clues how to choose $k^{+/-}$ and $\alpha^{+/-}$. Before providing explicit parameterization we first establish the unbiasedness of these estimators and analyze what influence $k^{+/-}$ and $\alpha^{+/-}$ may have on the variance of the AMCS estimators.

3.2 Unbiasedness

In order to analyze the AMCS estimators given by Eq. (3.1) and Eq. (3.2) we will introduce an additional random index $J \sim \text{Uniform}(\{1-N, ..., M-1\})$ which will be used to select a single point out of the sampled trajectory uniformly at random. Specifically, we define $X^{(J)} := \sum_{j=1-N}^{M-1} \mathbb{I}\{J = j\} X^{(j)}$. This random index is useful in establishing the unbiasedness of the AMCS estimator, in particular we will show that $\mathbb{E}\big[h(X^{(J)})\pi(X^{(J)})\big] = \mathbb{E}[\hat{\mu}_{AMCS}]$ and that $\mathbb{E}\big[h(X^{(J)})\pi(X^{(J)})\big] = \mu$. We begin by introducing some lemmas which will aid in proving the latter expression, it what follows we will make use of an Assumption 3.1 below. In practice this assumption can be achieved in a number ways, for example by forcing $\alpha^{+/-}$ to terminate the chain on any step with a fixed probability.

Assumption 3.1. *The Markov chains generated by $k^{+/-}$ and $\alpha^{+/-}$ are assumed to terminate eventually; i.e. $M < \infty$ and $N < \infty$ almost surely.*

In order to prove that $\mathbb{E}\big[h(X^{(J)})\pi(X^{(J)})\big] = \mu$ we will show that the procedure for sampling $X^{(J)}$ is identical to sampling from a *symmetric* Markov kernel k, that is a kernel

where $k(x, x') = k(x', x)$, and that any such procedure results in a unbiased estimator, as described in the following lemma.

Lemma 3.1. *Suppose* $Y \sim \pi(\cdot)$, $X \sim \pi_0(\cdot)$ *and* $X' \sim k(X, \cdot)$ *for symmetric Markov kernel* k, *that is* $k(x, x') = k(x', x)$. *Provided that* $\text{supp}(\pi) \subseteq \text{supp}(k \circ \pi_0)$ *it follows that*

$$\mathbb{E}\left[\frac{h(X')\pi(X')}{\pi_0(X)}\right] = \mathbb{E}[h(Y)] = \mu.$$

(The statement follows from Fubini's theorem; full proof is given in Appendix A.1.)

In order to show that $X^{(J)}$ is indeed sampled according to a symmetric Markov kernel we express the conditional density of $(X^{(-N)}, ..., X^{(M)})$ given $X^{(0)}$ in terms of $k^{+/-}$ and $\alpha^{+/-}$, that is

$$\begin{aligned}\gamma(x^{(-n)}, ..., x^{(m)}, n, m|x^{(0)}) := (1 - \alpha^+(x^{(m-1)}, x^{(m)}))k^+(x^{(m-1)}, x^{(m)}) \\ \times \prod_{j=1}^{m-1} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)}) \\ \times (1 - \alpha^-(x^{(1-n)}, x^{(-n)}))k^-(x^{(1-n)}, x^{(n)}) \\ \times \prod_{j=-1}^{1-n} \alpha^-(x^{(j+1)}, x^{(j)})k^-(x^{(j+1)}, x^{(j)}). \qquad (3.3)\end{aligned}$$

Here we are able to integrate out the acceptance variables $a^{(j)}$ since the conditional density is zero whenever we have invalid configurations, i.e. $a^{(-n)} \neq 0$, $a^{(m)} \neq 0$, or $a^{(i)} \neq 1$ (for $1 - n \leq i \leq m - 1$). We now observe that this is a valid conditional density and the density for a given set of points evaluates to the same value regardless of which of these points in chosen as the starting point $x^{(0)}$, as formalized in the following lemma.

Lemma 3.2. *Given jointly symmetric* (k^+, α^+) *and* (k^-, α^-), *and sequences* $(x^{(-n)}, ..., x^{(m)})$ *and* $(x'^{(-n')}, ..., x'^{(m')})$, *the density* γ *defined in Eq. (3.3) satisfies the following.*

i. $\int \sum_{n,m \geq 1} \gamma(x^{(-n)}, ..., x^{(m)}, n, m|x^{(0)}) dx_{-0}^{(-n:m)} = 1$

ii. $\gamma(x^{(-n)}, ..., x^{(m)}, n, m|x^{(0)}) = \gamma(x'^{(-n')}, ..., x'^{(m')}, n', m'|x'^{(0)})$

whenever $x^{(-n)} = x'^{(-n')}, x^{(1-n)} = x'^{(1-n')}, ..., x^{(m)} = x'^{(m')}$, *and* $m + n = m' + n'$ *where* $m, n, m', n' \geq 1$, *and where* $x_{-0}^{(-n:m)}$ *denotes all variables* $x^{(-n:m)}$ *but* $x^{(0)}$.
(Here (i) follows from Assumption 3.1 and (ii) from the definition of joint symmetry; the full proof is given in Appendix A.2.)

One remaining detail is to express the conditional density for only a single point ($X^{(J)}$) given $X^{(0)}$. For this we will make use of the following lemma:

Lemma 3.3. *Given random variables $(X_0, ..., X_L, L)$, where $X_i \in \mathbb{R}^d$, $L \in \mathbb{N}$, and $L \geq 2$, distributed according to some joint density γ and a random variable $J \sim \text{Uniform}(\{1, ..., L-1\})$, the variable $X_J = \sum_{j=0}^{l} \mathbb{I}\{J = j\} X_j$ has p.d.f. $p(x) = \sum_{l \geq 2} \frac{1}{l-1} \sum_{j=1}^{l-1} \gamma_j(x, l)$. Where γ_j is the j^{th} marginal density of γ, given by $\gamma_j(x_j, l) := \int \gamma(x_1, ..., x_l, l) dx_{1:l}^{-j}$.*
(proof given in Appendix A.3.)

With these components we are now able to state the first main result for this section which establishes the symmetry of the conditional density for $X^{(J)}|X^{(0)}$. However, before doing so we require some additional notation. Specifically by $x(n, m, x, j)$, for $n, m \geq 1$, $-n \leq j \leq m$, we denote the trajectory $(x^{(-n)}, ..., x^{(-1)}, x^{(1)}, x^{(m)})$ where $x^{(j)} = x$. Using Lemma 3.3, the conditional density $\gamma(x|x')$ may be written as

$$\gamma(x|x') = \sum_{l \geq 2} \frac{1}{l-1} \sum_{m=1}^{l-1} \sum_{j=m-l+1}^{m-1} \int \gamma(x(l-m, m, x, j), l-m, m | x^{(0)} = x') dx_{-\{j,0\}}^{(m-l:m)}, \tag{3.4}$$

where $\gamma(x(l-m, m, x, 0), l-m, m|x^{(0)} = x')$ is defined as zero whenever $x \neq x'$. Thanks ultimately to the symmetric properties of $\gamma(x(l-m, m, x, j), l-m, m|x^{(0)} = x')$ this conditional density also meets a critical symmetry property formalized in the following lemma.

Lemma 3.4. *Provided the density function γ defined in Eq. (3.3) satisfies the conditions in Lemma 3.2 the conditional density $\gamma(x'|x)$ defined in Eq. (3.4) satisfies*

$$\gamma(x'|x) = \gamma(x|x').$$

(The lemma follows from reordering sums in γ and deploying Lemma 3.2; full proof is given in Appendix A.4.)

Using these three lemmas we now formally establish the unbiasedness of AMCS in the following theorem.

Theorem 3.1. *Provided the transition kernels and acceptance functions satisfy the conditions of Lemma 3.2 and Lemma 3.4 for any $n > 0$ the AMCS procedure achieves*

$$\mathbb{E}[\hat{\mu}_{AMCS}^n] = \mu$$
$$\mathbb{E}\left[\hat{\zeta}_{AMCS}^n\right] = \zeta$$

Proof. Because $\hat{\mu}^n_{AMCS}$ is an empirical average of independent samples we need only show that each individual sample has the desired expectation. Recalling that $X^{(J)}$ is a single point sampled uniformly at random from a given trajectory $(X^{(-N)}, ..., X^{(M)})$ we first observe that

$$\begin{aligned}\mathbb{E}\left[\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}|(X^{(-N)}, ..., X^{(M)})\right]\right] \\ &= \mathbb{E}\left[\frac{\sum_{j=1-N}^{M-1} h(X^{(j)})\pi(X^{(j)})}{(M+N-1)\pi_0(X^{(0)})}\right] \\ &= \mathbb{E}[\hat{\mu}_{AMCS}]\,.\end{aligned}$$

It remains then only to show that $\mathbb{E}\left[\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\right] = \mu$. By defining our Markov kernel $k(x, x') = \gamma(x'|x)$ from Eq. (3.4), where $X^{(J)} \sim k(X^{(0)}, \cdot)$, it follows from Lemma 3.4 that $k(x, x') = k(x', x)$. As a result of this symmetry we may invoke Lemma 3.1 which gives the desired result. Noting that the identical steps hold for the estimator $\hat{\zeta}^n_{AMCS}$ as well, concludes the proof. □

3.3 Variance Analysis

Since the AMCS estimator is unbiased for any choice of jointly symmetric $k^{+/-}$ and $\alpha^{+/-}$ we now consider how these choices affect the MSE of the estimator. In the following developments we make use of the uniformly distributed index J as defined in the previous section. In particular we observe

$$\begin{aligned}v_{AMCS} &:= \mathbb{V}\left(\frac{\sum_{j=1-N}^{M-1} h(X^{(j)})\pi(X^{(j)})}{(M+N-1)\pi_0(X^{(0)})}\right) \\ &= \mathbb{V}\left(\mathbb{E}\left[\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle| X^{(-N)}, ..., X^{(M)}\right]\right),\end{aligned}$$

where the inner expectation is taken w.r.t. J. We are interested in the discrepancy between the above variance expression and that of vanilla importance sampling given by $v_{IS} := \mathbb{V}\left(\frac{h(X)\pi(X)}{\pi_0(X)}\right)$ for $X \sim \pi_0(\cdot)$. To relate these quantities we will first make the simplifying assumption that π_0 is uniform, so that the effects of the supplied proposal are negligible.[2]

[2] In practice π_0 need not be uniform, only that the density does not change significantly across any given trajectories.

Using this assumption and letting $k(x, x') = \gamma(x'|x)$ we observe

$$\begin{aligned}
\mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\right) &= \int\int \left(\frac{h(x')\pi(x')}{\pi_0(x)}\right)^2 k(x,x')\pi_0(x)dx'dx - \mu^2 \\
&= \int \left(\frac{h(x')\pi(x')}{\pi_0(x')}\right)^2 \pi_0(x')dx' \int k(x',x)dx - \mu^2 \\
&= \mathbb{V}\left(\frac{h(X)\pi(X)}{\pi_0(X)}\right) = v_{IS},
\end{aligned}$$

where we have used the symmetry properties of k (Lemma 3.4). This expression is essentially stating that if one were to actually use a uniformly drawn sample from each trajectory to estimate μ (as opposed to the average over all the points in a trajectory) the variance of the resulting estimator would be equal to v_{IS}. Next, using the law of total variance we see that

$$\begin{aligned}
v_{IS} &= \mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\right) \\
&= \mathbb{E}\left[\mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle|(X^{(-N)},...,X^{(M)})\right)\right] \\
&\quad + \mathbb{V}\left(\mathbb{E}\left[\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle|(X^{(-N)},...,X^{(M)})\right]\right) \\
&= \mathbb{E}\left[\mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle|(X^{(-N)},...,X^{(M)})\right)\right] + v_{AMCS}\,.
\end{aligned}$$

From this expression we derive the key result of this section, which we refer to as the *variance capture* identity:

$$v_{AMCS} = v_{IS} - \mathbb{E}\left[\mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle|X^{(-N)},...,X^{(M)}\right)\right]. \tag{3.5}$$

This identify shows that the variance of the AMCS estimator cannot be higher than the vanilla importance sampling estimator given the same number of samples (n) (i.e. ignoring the additional computational costs of AMCS). Additionally, the reduction in variance is determined entirely by the expected variance of the points inside a given trajectory under the uniform distribution. This observation motivates the use of *antithetic Markov chains*, in which the transition kernels $k^{+/-}$ are configured to head in opposite directions in the hopes of capturing more variability.

However, in order to get a fair understanding of the tradeoffs between AMCS and other approaches, it is important to consider the increased computational costs incurred by simulating the Markov chains. If we consider any Monte Carlo estimator which takes the

empirical average of an arbitrary sequence of i.i.d. random variables, say $X_1, ..., X_n$, then we know variance is given by $\frac{\mathbb{V}(X)}{n}$. If we also assume that this sampling procedure has a stochastic cost associated with each sample, denoted by the random variables $D_1, ..., D_n$ where $\delta := \mathbb{E}[D]$ also where $D_i \perp\!\!\!\perp X_i$, by fixing a computational budget $C >> \delta$, standard arguments for renewal reward processies indicate it will have a variance of approximately $\frac{\mathbb{V}(X)}{C/\delta} = \frac{\delta\mathbb{V}(X)}{C}$. Said simply, if technique A requires, on average, a factor of δ more computation per sample than technique B, then it must have a reduced variance by a factor of at least $1/\delta$ to be competitive. Plugging this forumla into the identity in Eq. (3.5) we can approximate that AMCS will offer a variance reduction whenever

$$\mathbb{E}\left[\mathbb{V}\left(\frac{h(X^{(J)})\pi(X^{(J)})}{\pi_0(X^{(0)})}\middle|(X^{(-N)}, ..., X^{(M)})\right)\right] > \frac{\delta - 1}{\delta} v_{IS},$$

where $\delta = \mathbb{E}[M + N + 1]$ gives the expected computational costs measured in terms of evaluations of the functions π and h. We say this value is approximate since the M and M are not technically independent of $(X^{(-N)}, ..., X^{(M)})$, but for large sample size the effects dependence will be negligible. That is, if the AMCS sampler requires 10 function evaluations each sample it will need to "capture" 90% of the variance of v_{IS} inside each trajectory, on average. It is clear then from this expression that the potential for savings drops of quickly as the per-sample computational costs increase. Interestingly, the above steps can be used to analyze a Monte Carlo estimator using antithetic variates as well, i.e. when $\delta = 2$, and also extends to the multiple variable setting. Also, an alternative analysis (for deterministic N,M) shows that the method would offer a variance reduction whenever $\sum_{i \neq j} \mathrm{Cov}\left(Z^{(i)}, Z^{(j)}\right) < 0$.

In the next section we explore explicit parameterizations for the Markov transitions and stopping rules that can be defined to capture variability while simultaneously keeping these costs in check.

3.4 Parameterization

In order to formulate concrete parameterizations for AMCS, it is first necessary to set out what properties of integrand we are hoping to exploit. In this section, keeping with the original motivations for AMCS, we consider parameterizations that are targeted toward multi-modal and peaked integrands. In these settings perhaps the most useful observation that can be made is that if the integrand is very near zero at a given point it is not likely to have large integrand values in its immediate vicinity. As a result, simulating Markov

chains in these areas is not likely to be worthwhile in terms of overall variance reduction. Conversely, if the integrand has some magnitude it has a much higher chance of being near a local mode. This observation motivates the *threshold acceptance function*

$$\alpha^{+/-}(x, x') = \begin{cases} 1, & \text{if } |f(x)| > \varepsilon \text{ and } |f(x')| > \varepsilon \\ 0, & \text{otherwise,} \end{cases} \tag{3.6}$$

where $\varepsilon > 0$ and $f(x)$ is some function of interest, likely the integrand ($h(x)\pi(x)$). An important aspect of this acceptance function is that if the first point sampled from π_0 is below threshold, the AMCS procedure can return immediately without evaluating the integrand at any neighboring points, therefore incurring no additional computational costs. That is to say, the sampler will perform exactly as well as a vanilla importance sampler in these regions and any variance reductions will therefore depend entirely on its behaviour in regions where the integrand is above this threshold.

In regards to transition kernels, a natural first choice is the *linear* Markov kernel densities

$$k^+(x, \cdot) = \mathcal{N}(x + v, \sigma^2 I),$$
$$k^-(x, \cdot) = \mathcal{N}(x - v, \sigma^2 I),$$

where $\mathcal{N}(\mu, \Sigma)$ denotes a multivariate normal distribution with mean μ and covariance Σ, $v \in \mathbb{R}^d$ is some fixed vector, and σ^2 a fixed variance parameter. In general these transitions should be parameterized so that $\sigma^2 << ||v||^2$ so that the resulting Markov chain will make consistent progress in one direction and typically experience more variability than, say, a normal random walk given the same number of steps.

For continuously differentiable integrands we can use the gradient to set the direction vector automatically giving rise to the *Langevin* Markov kernels

$$k^+(x, \cdot) = \mathcal{N}(x + \varepsilon \nabla f(x), \sigma^2 I),$$
$$k^-(x, \cdot) = \mathcal{N}(x - \varepsilon \nabla f(x), \sigma^2 I), \tag{3.7}$$

where $\varepsilon > 0$ is a fixed step size parameter. Since the gradient points in the direction of steepest ascent this choice seems ideal for capturing variability within a trajectory. Interestingly, this Markov kernel has been used extensively in the MCMC literature and, in fact, can be shown to produce Markov chain with stationary distribution f when $\sigma^2 = 2\varepsilon$ and $\varepsilon \to 0$ (Neal, 2011). However, when used in AMCS these moves can (and should) be parameterized so that $\varepsilon >> \sigma^2$, as a result the Markov chains look more like those generated from a steepest descent optimization algorithm than a MCMC diffusion.

One important concern with the Langevin kernel is that for nonlinear functions the transitions are not exactly symmetric. While this issue can be partially addressed by ensuring the gradient vector is normalized to length 1, exact joint symmetry (Definition 3.1) is best attained through the use of the *symmetrizing* acceptance functions

$$\alpha^{+}(x, x') = \min\left(\frac{k^{-}(x, x')}{k^{+}(x, x')}, 1\right),$$
$$\alpha^{-}(x', x) = \min\left(\frac{k^{+}(x', x)}{k^{-}(x', x)}, 1\right).$$

Note that multiple acceptance functions can be combined into a single function by taking the product.

Lastly, when taking gradient steps in either direction one can expect to eventually settle into a local mode or plateau. In these regions it continuing the chain will not capture any additional variation, and it is therefore beneficial to terminate it. This can be accomplished through the use of the *monotonic* acceptance functions

$$\alpha^{+}(x, x') = \begin{cases} 1, & \text{if } f(x) + \varepsilon < f(x') \\ 0, & \text{otherwise,} \end{cases}$$
$$\alpha^{-}(x, x') = \begin{cases} 1, & \text{if } f(x) - \varepsilon > f(x') \\ 0, & \text{otherwise,} \end{cases}$$

where $\varepsilon \geq 0$ is some fixed threshold. This acceptance function ensures that the chains make monotonic progress in either direction.

3.5 Experimental Evaluation

We now consider a number of empirical tests designed to evaluate our previous claims that the AMCS approach can reduce the statistical error of a vanilla importance sampling approach. Additionally, we aim to uncover whether these reductions are comparable to those offered by alternative state of the art Monte Carlo approaches. Specifically, we contrast the behaviour of an AMCS sampler with that of vanilla *importance sampling* (IS), *annealed importance sampling* (AIS), and *greedy importance sampling* (GIS) Southey et al. (2002). The previously unmentioned GIS approach is somewhat similar to the AMCS approach in terms of underlying mechanics since it uses a sequence of deterministic, axis-aligned, steepest ascent moves to augment a fixed proposal. In fact, the AMCS approach ultimately spawned from earlier attempts to extend the GIS approach to exploit continuous gradient information, as opposed to the expensive finite difference approximations. Interestingly, in the course of this work we observed that the AMCS acceptance functions could also be used

with the GIS approach with minimal effort. Indeed, the threshold acceptance function resulted in considerable improvements in a early testing. As a result, we added this improved GIS approach, dubbed GIS-A, to our suite of comparison approaches.

In contrasting these different approaches, careful consideration of the additional computational effort per sample is necessary. To account for these additional costs we measured the expected number of integrand evaluations per sample (denoted δ_M for method M) and compared methods using the *cost-adjusted variance* defined as $\bar{v}_M := \delta_M n \mathbb{V}(\hat{\mu}_M^n)$. Additionally, to ensure a meaningful comparison across experiments, we normalized this value by taking its ratio between the variance of the vanilla importance sampling approach to give the *relative cost-adjusted variance* given by $\bar{v}_M / \bar{v}_{IS}$. Here, a value of 0.5 indicates a 2x reduction in the number of integrand evaluations needed to attain the same error as an importance sampling estimator.[3]

For our comparisons we considered three different problem scenarios, first a synthetic problem with having numerous modes and taking on negative values, second a Bayesian k-means posterior, and finally a Bayesian posterior for a robot localization task.

3.5.1 Sin Function

We first consider a synthetic integration problem defined using a d-dimensional function $h(x) = \prod_{i=1}^d \sin(x_i)^{999}$ and density $\pi(x) = \mathcal{U}(x; 0, 3\pi)$, where $\mathcal{U}(x; a, b)$ denotes the multivariate uniform density on the interval $(a, b)^d$. At first glance it may seem that the exponent in h is a rather extreme choice, however, to give some perspective we note that a mode for this function is roughly the same size and shape as a normal density with $\sigma = 0.05$. Unsurprisingly, this integral poses challenges for numerical integration methods due to both its peaked landscape and large number (3^d) of separated modes. However, as it turns out, the most challenging aspect of this problem is the fact that the integrand takes on both positive and negative values. As a direct consequence the most effective SMCS approaches, where one targets a sampler toward the un-normalized distribution $h\pi$, cannot be deployed, and since π is a simple uniform distribution it is obviously not necessary, or worthwhile, to target a complex sampler toward this distribution.

One alternative approach for such problems is the so-called *harmonic mean* (HM) method which involves first simulating a sequence of points $X^{(1)}, ..., X^{(n)}$ from $|h|\pi$, using a MCMC approach, then approximating $\mathcal{I}$ using the weighted importance sampling

[3]Note that in our analysis we do not apply additional costs for gradient evaluations since, in most settings, computations of $h(x)$ and $\nabla h(x)$ typically share the same sub-computations which can be cached and reused. Gradients with respect to mini-batches may also be used.

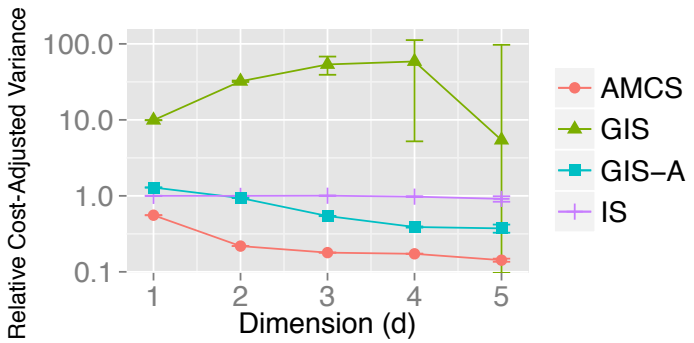

Figure 3.3: Cost-adjusted variance (log-scale) for the various methods on the $\sin(x)^{999}$ function. GIS referes to the original greedy importance sampling approach and GIS-A the extended version using the threshold acceptance function.

estimator (2.3) (see Newton and Raftery (1994)). The resulting estimator is consistent but often has extremely high (or infinite) variance even when the MCMC chain is able to mix efficiently. Despite extensive experimentation we were not able to find a parameter setting for which the HM estimator was not able to produce even remotely accurate estimates for $d > 2$. Ultimately, there are very few Monte Carlo approaches that can offer a fair comparison on integrands having both positive and negative values and, as a result, we consider ACMS alongside the GIS and IS approaches only.

For these comparisons the AMCS approach was parameterized with a proposal density $\pi_0 = \pi$, *Langevin* Markov kernels with $f = h$, step-size $\varepsilon = 0.03$, and $\sigma^2 = 3\text{E}^{-5}$, and *symmetrizing* and *monotonic* acceptance functions (with threshold $\varepsilon = 0$). Additionally, we used the *threshold* acceptance functions using the squared 2-norm of the gradient vector, that is $f = ||\nabla h||^2$, and a threshold value of $\varepsilon = 1\text{E}^{-15}$. For the GIS approach we used the same step-size ($\varepsilon = 0.03$) and, for the GIS-A variant, the same acceptance function.

The relative performance of each of these methods, as the dimensionality of the problem is increased, is plotted in Fig. 3.3. Again, we consider the cost-adjusted variance relative to the vanilla importance sampling approach variance which uses direct Monte-Carlo sampling from π (calculated analytically). The points plotted under the 'IS' heading are those computed through simulation and confirm the accuracy of the calculation and give some indication of the statistical error for this simulation. As mentioned previously, the simulations for the harmonic mean method were not informative as they were either off the chart where too noisy to permit empirical evaluation. As for the GIS method, the results clearly indicate that incorporating the threshold acceptance function (GIS-A) can lead to a significant increase in performance and, in this example, yields up to a 150-fold improvement over the original method. This is a welcome improvement since the original method performed

much worse than even simple importance sampling especially as the dimensionality of the integrand was increased. Ultimately, however, the AMCS approach is clearly the most effective approach for this task as it consistently outperformed all other approaches offering up to an 8x improvement over direct sampling. Moreover, the relative performance of the AMCS approach seemed to improve as the dimensionality of the intergrand was increased.

3.5.2 Bayesian k-mixture Model

In our next experiment we consider the task of approximating the normalization constant (ζ), or model evidence, for a Bayesian k-mixture model. Specifically, we define a generative model with k uniformly weighted multivariate normal distributions in $\mathbb{R}^d$ with fixed diagonal covariance matrices $\Sigma_i = \frac{i}{20} I$ for $i = \{1, ..., k\}$. The unobserved latent variables for this model are the means for each component $\mu_i \in \mathbb{R}^d$ which are assumed to be drawn from a multivariate normal prior with mean zero and identity covariance. Given n samples, $y_1, ..., y_n$, from the true underlying model, the model evidence is defined as the integration of the un-normalized posterior

$$\zeta = \int \prod_{i=1}^{n} \mathcal{L}(\mu_1, ..., \mu_k | y_i) p(\mu_1, ..., \mu_k) d\mu_1 ... d\mu_k,$$

where the likelihood function is given by $\mathcal{L}(\mu_1, ..., \mu_k | y_i) = \sum_{j=1}^{k} \frac{1}{k} \mathcal{N}(y_i; \mu_j, \Sigma_j)$ and the prior density the standard normal $p(\mu_1, ..., \mu_k) = \mathcal{N}([\mu_1, ..., \mu_k]; 0, I)$, where $[\mu_1, ..., \mu_k]$ denotes a dk-dimensional vector of "stacked" μ_i vectors. To use the same notation as previous sections we may write $\hat{\pi}(x) = \prod_{i=1}^{n} \mathcal{L}(x | y_i) p(x)$, where $x = [\mu_1, ..., \mu_k]$.

For this task standard SMCS approaches can be applied in a straightforward manner though they do require some preliminary parameter tuning. After some experimentation we found that the AIS approach performed well with 150 annealing distributions set using the "power of 4" heuristic suggested by Kuss and Rasmussen (2005), i.e. $\beta_i = ((150 - i)/150)^4$. Each annealing stage used 3 MCMC transitions, here we evaluate both slice sampling Neal (2003) and Hamiltonian transitions Neal (2011). The Hamiltonian moves were tuned to achieve a accept/reject rate of about 80% which resulted in a step-size parameter of 0.003 and 5 leapfrog steps. Additionally, for AIS and the remaining methods we use the prior as the proposal density, $\pi_0 = p$, and the posterior as the target.

For AMCS we used Langevin local moves with monotonic, symmetrizing, and threshold acceptance functions. For the Langevin moves we used a step-size parameter $\varepsilon = 0.015$ and $\sigma^2 = 3\text{E}^{-5}$ which, again, was set though some manual tuning. The threshold acceptance functions were configured using a preliminary sampling approach. In particular, let-

ting $f = \hat{\pi}$ we set the threshold parameter to a value that accepted roughly 1.5% of the data points on a small sub-sample (2000 points). These points were not used in the final estimate but in practice they can be incorporated without adverse effects. For the GIS approach we used step-size parameter $\varepsilon = 0.015$, also, we experimented the modified version (GIS-A) which used the same threshold acceptance function as the AMCS approach.

The results for this problem are shown in Fig. 3.4 as the number of "training" points (n), the dimensionality of these points (d), and number of mixture components (k) are altered. For each of these different settings the parameters for the sampling approaches remain fixed. Simulations were run for a period of 8 hours for each method and each setting of d, n, and k giving a total running time of 106 CPU days running on a cluster with 2.66GHz processors. However, even in this time many of the methods were not able to return a meaningful estimate after execution, these results are therefore omitted from the figure.

Looking at the relative variances plotted in Fig. 3.4 it is immediately clear from these simulations that GIS (both variants) and AIS with Hamiltonian moves (AIS-HAM) are simply not effective for this task as they perform several orders of magnitude worse than even vanilla IS. The AIS approach with slice sampling moves (AIS-SS) and the AMCS approach, however, exhibit much more interesting behaviour. In particular, the experiments indicate that AIS-SS can offer tremendous savings over vanilla IS (and sometimes AMCS) for higher dimensional problems and problems with more training samples. However, this advantage seems to comes at a price, as the method performed up to 10-20x worse than vanilla IS in other cases, essentially where the posterior distribution was not so challenging for IS. AMCS, on the other hand, was considerably more robust to changes in the target, since for each setting it performed at least as well as vanilla importance sampling. Additionally, in the more challenging settings it offered a considerable advantage over IS.

In summary, depending on the problem at hand, and the practitioner's appetite for risk, the most appropriate approach for this particular problem is either AMCS or AIS-SS. In many cases however, the practitioner may be interested in a large set of potential problem settings where it is not possible to determine which method, and parameter settings, are most appropriate for each case. In such scenarios it may be worthwhile to consider an *adaptive* algorithm to select from a set of fixed approaches automatically. In the remaining chapters of this thesis we consider adaptive strategies precisely of this form.

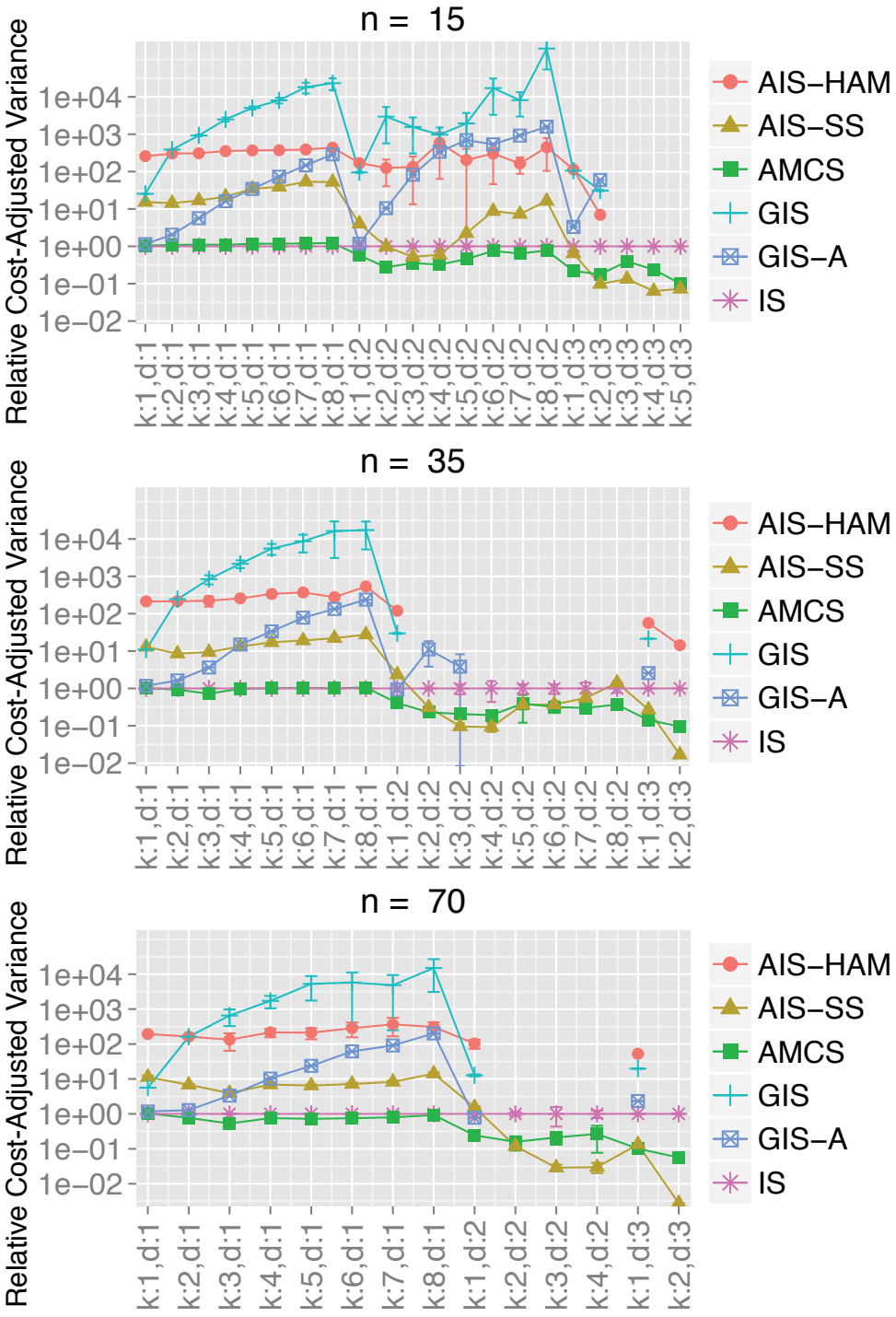

Figure 3.4: Cost-adjusted variance (log scale) for the different approaches on the Bayesian k-means task. Missing data points are due to the fact that trials where the final estimate (empirical mean) is incorrect by a factor of 2 or greater are automatically removed. From left to right the three plots indicate performance on the same problem but with an increasing number of observed training samples 15, 35, and 70 respectively.

3.5.3 Problem 3: Robot Localization

Our final simulations are centered around the approximation of the normalization constant for a Bayesian posterior for the (simulated) *kidnapped robot problem* (Thrun et al., 2005). In this setting an autonomous robot is placed at an unknown location and must recover its position using relative sensors, such as a laser range finder, and a known map. This posterior distribution is notoriously difficult to work with when the sensors are highly accurate since this produces a highly peaked distribution; a phenomenon referred to as *the curse of accurate sensors*. Here, we assume the prior distribution over the robot's (x,y) position and orientation, denoted $x \in \mathbb{R}^3$, is a uniform distribution.

In our simulations the robot's observations are akin to those produced by a laser range finder returning distance measurements at n positions spaced evenly in a 360° field of view (see Fig. 3.5). The sensor model for each individual sensor, that is, the likelihood of observing a measurement y given the true ray-traced distance from position x: $d(x)$, is given by the mixture $\mathcal{L}(y|d(x)) = 0.95\mathcal{N}(y; d(x), \sigma^2) + 0.05\mathcal{U}(y; 0, M)$, where $\sigma^2 = 4$cm and the maximum ray length $M = 25$m.[4] This sensor model is used commonly in the literature (see Thrun et al. (2005)) and is meant to capture the noise inherent in laser measurements (normal distribution) as well as moving obstacles or failed measurements (uniform distribution). Given a set of observed measurements $y_1, ..., y_n$ then, we have the un-normalized posterior distribution $\hat{\pi}(x) = \prod_{i=1}^{n} \mathcal{L}(y_i|d_i(x))p(x)$, where p denotes the density of the uniform prior.

The posterior distribution for a fixed observation and orientation are illustrated in Fig. 3.5, the true robot position and laser measurements on the left and the log-likelihood on the right, additionally a similar 3d plot was shown earlier in Fig. 3.1. These plots help to illustrate the highly multimodal and peaked lanscape which poses challenges for standard integration approaches. Additionally, the fact that integrand values require an expensive ray-tracing procedure to compute underscores the importance of efficient sampling routines. Also, due to the sharp map edges and properties of the observation model, the posterior distribution is highly non-continuous and non-differentiable. This prevents the use of gradient-based Markov transition (for AMCS and AIS) and severely limits the effectiveness of annealing the target density.

Again, through some manual tuning we found that a suitable parameterization for AIS was to use 100 annealing distributions each featuring 3 Metropolis-Hastings MCMC steps

[4] Measurements assume that the map (Fig. 3.5) is 10x10 meters.

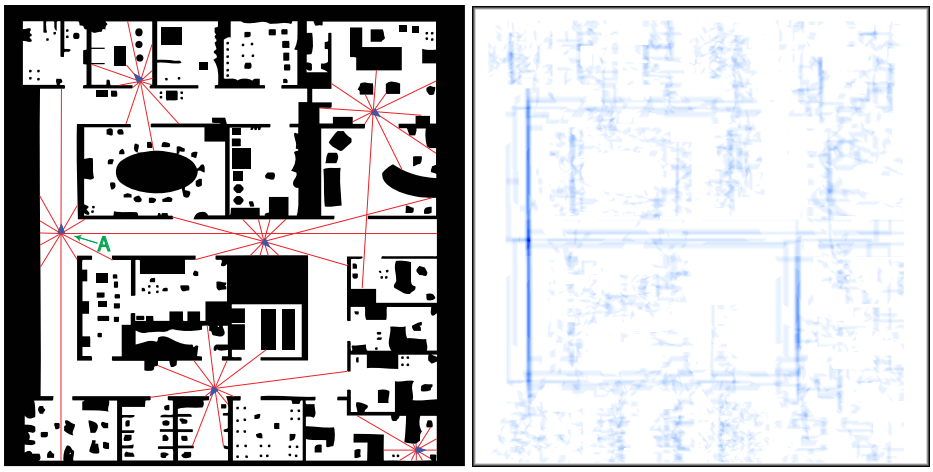

Figure 3.5: Left, the map used for the robot simulator with 6 different robot poses and corresponding laser measurements (for $n = 12$). Right, a 2d image where %blue is proportional to the log-likelihood function using the observations shown at position 'A', here pixel locations correspond to robot (x, y) position while the orientation remains fixed.

with proposal $q(x, x') = \mathcal{N}(x'; x, \sigma^2 I)$ with $\sigma^2 = 4$cm. For AMCS, we used the prior as a proposal density, *linear* Markov kernels with $v = [2\text{cm}, 2\text{cm}, 0.2\text{cm}]$ and $\sigma^2 = 2\text{E}^{-3}$cm and threshold acceptance function with threshold set to be larger than 4% of points on a 2000 point sub-sample. For GIS, we used the same proposal, step-sizes, and (for GIS-A) the same threshold acceptance function as AMCS.

The relative error rates for the different sampling approaches for 6 different positions (see Fig. 3.5) and 3 different laser configurations, $n = 12, 18, 24$, are shown in Fig. 3.6. Unlike the previous task, the results here are very straightforward and indicate that AMCS consistently offers a 8-10x improvement over vanilla importance sampling. Again, the the GIS approach can be significantly improved through the addition of threshold acceptance functions but even so, it is only marginally better than vanilla IS. Lastly, it is clear from these results that AIS is simply not an effective approach for this task as it is roughly 10x less efficient than simple IS and 100x less efficient than AMCS. This is primarily due to the fact that the unmodified proposal density has some reasonable chance of landing in or near an integrand peak. Consequently, taking a large number of MCMC transitions is simply not a cost-effective means to improving the proposal, this detail exacerbated by landscape of the posterior distribution which inhibits efficient MCMC mixing.

3.5.4 Discussion

In this chapter we detailed the antithetic Markov chain sampling approach can be seen as a unique way to extend the importance sampling (or antithetic variates) approach through

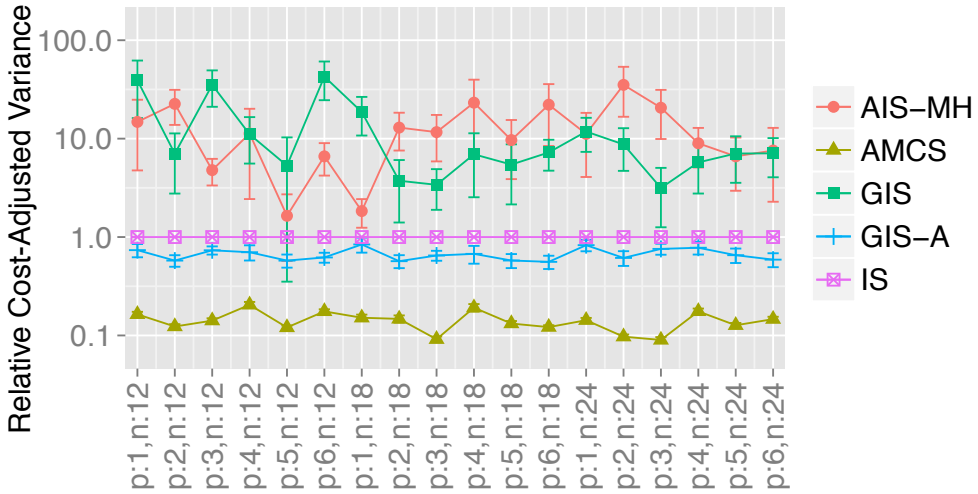

Figure 3.6: Relative cost-adjusted variance for the different approaches on the robot localization task for 6 different positions (p) and 3 different laser configurations (n = #laser readings).

the use of local Markov chains. The approach can be parameterized in a unique number of ways to allow the sampler to exploit specific aspects of the target integrand such as differentiability and the existence of local modes. The experimental comparisons in Section 3.5 demonstrate that the approach, when appropriately configured, can result in computational savings that exceed the state of the art on non-trivial problems. Additionally, we previously alluded to a handful of interesting limitations and potential extensions to this approach for which we now elaborate.

One interesting shortcoming of the AMCS approach is the fact that when using Langevin kernels the acceptance rates begin to drop off as the Markov chain approaches an optimum. This effect is magnified as the dimensionality or curvature of the integrand is increased. Upon closer examination we find that this effect results from a combination of two factors. First, low acceptance rates are an unavoidable consequence of the way gradient-based moves concentrate around an optimum, said simply, the acceptance rate is essentially avoiding "over counting" the integrand values within a given region. The second contributing factor is related to the use of a fixed step size and variance in the Langevin moves. For example, if we first consider a single transition toward an optimum, $k^{+}(x, x')$, then consider a transition in the other direction, $k^{-}(x', x)$, we often find that the gradient is pointing in a very different direction; which leads to a low acceptance rate. With this effect in mind, one natural avenue for improvement is to adjust the step-size and variance of the Langevin moves depending on the second derivative information of the target function. That is, if the

gradient is changing very quickly the amount of noise and the step sizes should be adjusted depending on the direction of change.

Perhaps the most critical limitation of the AMCS is the number of free parameters that must be set by the practitioner before any sampling occurs. In the empirical evaluations of this approach the parameters were tuned manually through the use of preliminary experiments. Of course, in many settings it is either cost-prohibitive, or not possible, to have a human in the loop when once the sampler is deployed. A more robust approach to parameter tuning would be to *adapt* the parameters in an online fashion according to past simulations, as is done in many of the adaptive approaches reviewed in Chapter 2. However, unlike SMCS the AMCS does not utilize a population of samples at run time. While this is generally seen as a beneficial aspect it does preclude the ability to alter the parameters of the algorithm using populations, as is done in both the population Monte Carlo and adaptive SMCS approaches. However, the strategy taken by the online learning-based strategies used in the adaptive stratified sampling approach of Carpentier and Munos (2011) hints at a more suitable way of conducting adaptation in this space. Indeed, through the subsequent chapters we explore ways in which multi-armed bandit methods can be used to adapt AMCS and many other Monte Carlo integration methods.

Chapter 4

Adaptive Monte Carlo via Bandit Allocation

"My last piece of advice to the degenerate slot player who thinks he can beat the one-armed bandit consists of four little words: 'It can't be done.' "

– John Scarne

In the previous chapter we remarked that having some way of automatically tuning parameters would greatly improve the practical significance of the antithetic Markov chain sampling approach. However, AMCS differs from many of the existing Monte Carlo approaches that enjoy adaptive counterparts, therefore formulating an adaptive scheme is more involved. In particular, the AMCS formulation does not permit a straightforward expression for the variance of the estimator as does, for example, parametric importance sampling (Eq. (2.4)). Additionally, the approach does not employ resampling strategies which prevents one from adopting the techniques used in the sequential Monte Carlo literature. In considering alternative approaches we observe that there are actually a number of existing Monte Carlo approaches that are equally challenging to parameterize. As a result, in this chapter, we explore a general strategy that can be used to construct adaptive variants for many of these Monte Carlo methods.

Specifically, we consider an adaptive Monte Carlo framework, *learning to select Monte Carlo samplers*, in which computation (samples) is allocated sequentially between a set of unbiased Monte Carlo estimators with the goal of minimizing the expected squared error (MSE) of a final combined estimate (Neufeld et al., 2014). Unlike many of the existing adaptive Monte Carlo methods this method is not tied to any specific sampling approach and can therefore be used to find not only the best parameter setting but the best underlying Monte Carlo approach as well. In order effectively manage this sequential allocation we

make use of techniques developed in the *online learning* literature, specifically *stochastic multi-armed bandit* methods (Bubeck and Cesa-Bianchi, 2012; Cesa-Bianchi and Lugosi, 2006; Auer et al., 2002a). Indeed, the primary contribution of this chapter is a reduction of this adaptive Monte Carlo formulation to the stochastic bandit setting. As a consequence, existing bandit algorithms can be immediately applied to achieve new results for adaptive Monte Carlo estimation. In addition to providing a nontrivial practical advantage, these strategies permit a broader theoretical analysis than traditional *adaptive* Monte Carlo strategies which ultimately leads to strong finite-time performance guarantees.

This work is closely related, and complementary, to the work on adaptive stratified sampling reviewed in Section 2.5, in which samples were allocated between fixed strata to achieve MSE-regret bounds relative to the best allocation in hindsight. The method proposed in this chapter, however, can be applied more broadly to any set of base estimation strategies and potentially even in combination with these approaches. Before formalizing the adaptive Monte Carlo problem setting we provide some background on the stochastic multi-armed bandit setting which will set the stage for the subsequent developments.

4.1 Background on Bandit Problems

The multi-armed bandit (MAB) problem is a sequential allocation task where an agent must choose an action[1] at each step to maximize its long term payoff, when only the payoff of the selected action can be observed (Cesa-Bianchi and Lugosi, 2006; Bubeck and Cesa-Bianchi, 2012). In the *stochastic* MAB problem (Robbins, 1952) the payoff for each action $k \in \{1, \dots, K\}$[2] is assumed to be generated independently and identically (i.i.d.) from a fixed but unknown distribution ν_k. The performance of an allocation policy can then by analyzed by defining the *cumulative regret* for any sequence of n actions, given by

$$R_n := \max_{1 \le k \le K} \mathbb{E}\Big[\sum_{t=1}^{n} X_{k,t} - \sum_{t=1}^{n} X_{I_t, T_{I_t}(t)}\Big], \tag{4.1}$$

where $X_{k,t} \in \mathbb{R}$ is the random variable giving the t-th payoff of action k, $I_t \in \{1, \dots, K\}$ denotes the action taken by the policy at time-step t, and $T_k(t) := \sum_{s=1}^{t} \mathbb{I}\{I_s = k\}$ denotes the number of times action k is chosen by the policy up to time t. The objective of the agent is to maximize the total payoff, or equivalently to minimize the cumulative regret. By

[1] Alternatively we may refer to the different actions as "arms".

[2] To remain consistent with existing literature we deviate from our regular notation and denote the fixed (non-random) number of arms with the uppercase K.

rearranging (4.1) and conditioning, the regret can be rewritten

$$R_n = \sum_{k=1}^{K} \mathbb{E}[T_k(n)](\mu^* - \mu_k), \tag{4.2}$$

where $\mu_k := \mathbb{E}[X_{k,t}]$ and $\mu^* := \max_{j=1,\dots,K} \mu_j$.

The analysis of the stochastic MAB problem was pioneered by Lai and Robbins (1985) who showed that, when the payoff distributions are defined by a single parameter, the asymptotic regret of any sub-polynomially consistent policy (i.e., a policy that selects non-optimal actions only sub-polynomially many times in the time horizon) is lower bounded as $\Omega(\log n)$. In particular, for Bernoulli payoffs we have

$$\liminf_{n\to\infty} \frac{R_n}{\log n} \geq \sum_{k:\Delta_k>0} \frac{\Delta_k}{\mathrm{KL}(\mu_k, \mu^*)}, \tag{4.3}$$

where $\Delta_k := \mu^* - \mu_k$ and $\mathrm{KL}\,(p, q) := p \log(p/q) + (1-p) \log(\frac{1-p}{1-q})$ for $p, q \in [0, 1]$. In the same work Lai and Robbins also presented an algorithm based on upper confidence bounds (UCB), which achieves a regret asymptotically matching the lower bound (for certain parametric distributions).

Later, Auer et al. (2002a) proposed UCB1 (Algorithm 5), which broadens the practical application of UCB by dropping the requirement that payoff distributions fit a particular parametric form. Instead, we need only make a weaker assumption that the rewards are bounded; in particular, we let $X_{k,t} \in [0, 1]$. Auer et al. proved that, for any finite number of actions n, UCB1's regret is bounded by

$$R_n \leq \sum_{k:\Delta_k>0} \frac{8 \log n}{\Delta_k} + \Big(1 + \frac{\pi^2}{3}\Big)\Delta_k. \tag{4.4}$$

Algorithm 5 UCB1 (Auer et al., 2002a)

1: **for** $k \in \{1, \dots, K\}$
2: Play k, observe $X_{k,1}$, set $\hat{\mu}_{k,1} := X_{k,1}$; $T_k(1) := 1$.
3: **end for**
4: **for** $t \in \{K+1, K+2, \dots\}$
5: Play action k that maximizes $\hat{\mu}_{j,t-1} + \sqrt{\frac{2 \log t}{T_j(t-1)}}$; set $T_k(t) = T_k(t-1) + 1$ and $T_j(t) = T_j(t-1)$ for $j \neq k$, observe payoff $X_{k,T_k(t)}$, and compute $\hat{\mu}_{k,t} = (1 - 1/T_k(t))\hat{\mu}_{k,t-1} + X_{k,T_k(t)}/T_k(t)$.
6: **end for**

Various improvements of the UCB1 algorithm have since been proposed. One approach of particular interest is the UCB-V algorithm (Audibert et al., 2009), which takes the empirical variances into account when constructing confidence bounds. Specifically, the UCB-V

bound is given by

$$\hat{\mu}_{k,t} + \sqrt{\frac{2\hat{\sigma}^2_{k,T_k(t-1)}\mathcal{E}_{T_k(t-1),t}}{T_k(t-1)}} + c\frac{3\mathcal{E}_{T_k(t-1),t}}{T_k(t-1)}, \tag{4.5}$$

where $\hat{\sigma}^2_{k,s}$ denote the sample variance of arm k after s samples, $c > 0$, and $\mathcal{E}_{s,t}$ is an *exploration function* required to be a non-decreasing function of s or t (typically $\mathcal{E}_{s,t} = \gamma \log(t)$ for a fixed constant $\gamma > 0$). The UCB-V procedure can then be constructed by substituting the above confidence bound into Algorithm 5, which yields a regret bound that scales with the true variance of each arm

$$R_n \leq c_\gamma \sum_{k:\Delta_k>0} \left(\frac{\sigma^2_k}{\Delta_k} + 2\right) \log n. \tag{4.6}$$

Here c_γ is a constant relating to γ and c and $\sigma^2_k = \mathbb{V}(X_{k,t})$. In the worst case, when $\sigma^2_k = 1/4$ and $\Delta_k = 1/2$, this bound is slightly worse than UCB1's bound; however, it is usually better in practice, particularly if some k has small Δ_k and σ^2_k.

A more recent algorithm is KL-UCB (Cappé et al., 2013), where the confidence bound for arm k is based on solving

$$\sup\left\{\mu : \mathrm{KL}\left(\hat{\mu}_{k,t}, \mu\right) \leq \frac{f(t)}{T_k(t)}\right\}, \tag{4.7}$$

for a chosen increasing function $f(\cdot)$, which can be solved efficiently since $\mathrm{KL}\left(p, \cdot\right)$ is smooth and increasing on $[p, 1]$. By choosing $f(t) = \log(t) + 3\log\log(t)$ for $t \geq 3$ (and $f(1)=f(2)=f(3)$), KL-UCB achieves a regret bound

$$R_n \leq \sum_{k:\Delta_k>0} \left(\frac{\Delta_k}{\mathrm{KL}\left(\mu_k, \mu^*\right)}\right) \log n + O(\sqrt{\log(n)}), \tag{4.8}$$

for $n \geq 3$, with explicit constants for the "higher order" terms (Cappé et al., 2013, Corollary 1). Apart from the higher order terms, this bound matches the lower bound (4.3). In general, KL-UCB is expected to be better than UCB-V except for large sample sizes and small variances. Note that, given any set of UCB algorithms, one can apply the tightest upper confidence from the set, via the union bound, at the price of a small additional constant in the regret.

Another approach that has received significant recent interest is Thompson sampling (TS) (Thompson, 1933): a Bayesian method where actions are chosen randomly in proportion to the posterior probability that their mean payoff is optimal. TS is known to outperform UCB-variants when payoffs are Bernoulli distributed (Chapelle and Li, 2011; May and

Leslie, 2011). Indeed, the finite time regret of TS under Bernoulli payoff distributions closely matches the lower bound (4.3) (Kaufmann et al., 2012):

$$R_n \leq (1+\varepsilon) \sum_{k:\Delta_k>0} \frac{\Delta_k(\log(n)+\log\log(n))}{\mathrm{KL}\left(\mu_k, \mu^*\right)} + C(\varepsilon, \mu_{1:K}),$$

for every $\varepsilon > 0$, where C is a problem-dependant constant. However, in light of the non-parametric characteristics of the previous bandit approaches the assumption on Bernoulli distributed rewards seems quite strong; indeed not possible to have Bernoulli distributed payoffs with the same mean and different variances. Fortunately, a more general version of Thompson sampling can be formulated through a simple resampling step (Agrawal and Goyal, 2012), this non-parametric Thompson sampling variant is given in Algorithm 6 and has been shown to obtain

$$R_n \leq \Big(\sum_{k:\Delta_k>0} \frac{1}{\Delta_k^2} \Big)^2 \log(n). \tag{4.9}$$

Algorithm 6 Thompson Sampling (Agrawal and Goyal, 2012)

Require: Prior parameters α and β
Initialize: $S_{1:K}(0) := 0$, $F_{1:K}(0) := 0$
for $t \in \{1, 2, \ldots\}$
  Sample $\theta_{t,k} \sim \mathcal{B}(S_k(t-1)+\alpha, F_k(t-1)+\beta)$, $k = 1...K$
  Play $k = \arg\max_{j=1,\ldots,K} \theta_{t,j}$; observe $X_t \in [0,1]$
  Sample $\hat{X}_t \sim \text{Bernoulli}(X_t)$
  if $\hat{X}_t = 1$ **then** set $S_k(t) = S_k(t-1) + 1$ **else** set $F_k(t) = F_k(t) + 1$
end for

There are of course a large number of alternative bandit methods that we do not consider in detail, each having unique advantages and limitations. More notable variants include the UCB-tuned algorithm of Auer et al. (2002a) which can exploit knowledge of the variance in a similar fashion to UCB-V, though this work did not include theoretical bounds it has been shown to sometimes outperform UCB-V (Cappé et al., 2013). Another high-profile upper confidence bound algorithm is the MOSS approach of Audibert and Bubeck (2010), which exploits knowledge of the time horizon to tweak the standard UCB bound. This approach is known to work well in practice and has much stronger theoretical guarantees.

4.2 Adaptive Monte Carlo Setup

We now formalize the general adaptive Monte Carlo setting. In particular, we assume we are given a finite number of Monte Carlo samplers (arms) where each base sampler $k \in \{1, .., K\}$ produces an i.i.d. sequence of real-valued random variables $\{X_{k,t}\}_{(t=1,2,\ldots)}$

having the same unknown mean, that is, $\mathbb{E}[X_k] = \mu$. In practice, such samplers could include any unbiased method and/or variance reduction technique, such as unique instantiations of importance sampling, sequential Monte Carlo, antithetic variates, or control variates (Robert and Casella, 2005). Critically, we do not assume to have any prior knowledge regarding the variance of each respective sampler. We assume also that drawing a sample from each sampler takes constant time, hence the samplers differ only in terms of how fast their respective sample means $\hat{\mu}_{k,n} = \frac{1}{n}\sum_{t=1}^{n} X_{k,t}$ converge to μ. The goal is to design a *sequential estimation procedure* that works in discrete time steps: For each round $t = 1, 2, \ldots$, based on the previous observations, the procedure selects one sampler $I_t \in \{1, \ldots, K\}$, whose observation is used by an outer procedure to update the estimate $\hat{\mu}_t \in \mathbb{R}$ based on the values observed so far.

As in the previous chapters, we will evaluate the accuracy of our final estimate using the mean-squared error (MSE). That is, we define the loss of the sequential method $\mathcal{A}$ at the end of round n by $L_n(\mathcal{A}) = \mathbb{E}\big[(\hat{\mu}_n - \mu)^2\big]$. A reasonable goal is to then compare the loss, $L_{k,n} = \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu)^2\big]$, of each base sampler to the loss of $\mathcal{A}$. In particular, we will evaluate the performance of $\mathcal{A}$ by the (normalized) regret

$$R_n(\mathcal{A}) = n^2\Big(L_n(\mathcal{A}) - \min_{1\le k\le K} L_{k,n}\Big)\,, \tag{4.10}$$

which measures the loss of $\mathcal{A}$ with respect to the best possible estimator. Implicit in this definition is the assumption that $\mathcal{A}$'s time to select the next sampler is negligible compared to the time to draw an observation. Note also that the excess loss is multiplied by n^2, which ensures that, in standard settings, when $L_{k,n} \propto 1/n$, a sublinear regret (i.e., $|R_n(\mathcal{A})|/n \to 0$ as $n \to \infty$) implies that the loss of $\mathcal{A}$ asymptotically matches that of the best estimator. Therefore, this normalization is meant to align the regret definition with regret notions in other online problems (like bandits), where a sublinear regret means "learning". Also note that a sublinear bound on the regret then captures how fast $L_n(\mathcal{A}) - \min_{1\le k\le K} L_{k,n}$ converges to zero.

In what follows we will adopt a simple strategy for combining the values returned from the base samplers, specifically, that $\mathcal{A}$ returns the (unweighted) average of all samples as the estimate $\hat{\mu}_n$. A more sophisticated approach might be to weight each of these samples inversely proportional to their respective (sample) variances. However, this approach is prone to overweighting some estimators due to statistical errors in the sample variances. Additionally, if the adaptive procedure can quickly identify and ignore highly suboptimal arms the savings from the weighted estimator will diminish over time. Nonetheless, this

weighting question come back into the fold when we consider extensions to this framework as well as empirical studies; so much in fact, that we devote Chapter 6 entirely to better understanding this issue.

4.3 Reduction to Stochastic Bandits

We now proceed with the technical analysis that will establish the equivalence between the stochastic multi-armed bandit setting and the adaptive Monte Carlo setting described above. Our main assumption in this section will be the following:

Assumption 4.1. *Each sampler produces a sequence of i.i.d. random observations with common mean μ and finite variance; values from different samplers are independent.*

Let ψ_k denote the distribution of samples from sampler k. Note that $\Psi = (\psi_k)_{1 \le k \le K}$ completely determines the sequential estimation problem. Recalling that $\sigma_k^2 = \mathbb{V}(X_k)$ we define $\sigma_{k^*}^2 := \min_{1 \le k \le K} \sigma_k^2$. Furthermore, since the sampled values are i.i.d. we observe $L_{k,t} = \sigma_k^2/t$, hence $\min_{1 \le k \le K} L_{k,t} = \sigma_{k^*}^2/t$. We may then state the first main result of this chapter.

Theorem 4.1 (Regret Identity)**.** *Consider K samplers for which Assumption 4.1 holds, and let $\mathcal{A}$ be an arbitrary allocation procedure. Then, for any $n \ge 1$, the MSE-regret of the estimation procedure $\mathcal{A}^{\text{avg}}$, estimating μ using the sample-mean of the observations obtained by $\mathcal{A}$, satisfies*

$$R_n(\mathcal{A}^{\text{avg}}) = \sum_{k=1}^{K} \mathbb{E}[T_k(n)] \, (\sigma_k^2 - \sigma_{k^*}^2) \, . \tag{4.11}$$

The proof is relatively straightforward and follows from a two key lemmas, which we detail below. Recalling that $I_t \in \{1, \dots, K\}$ denotes the choice that $\mathcal{A}$ made at the beginning of round t and that $T_k(t) = \sum_{s=1}^{t} \mathbb{I}\{I_s = k\}$. The observation at the end of round t is $Y_t = X_{I_t, T_{I_t}(t)}$ and the cumulative sum of returns after n rounds for arm k is $S_{k,n} = \sum_{t=1}^{T_k(n)} X_{k,t}$. Likewise, $S_n = \sum_{t=1}^{n} Y_t = \sum_{k=1}^{K} S_{k,n}$. By definition, the estimate after n rounds is

$$\hat{\mu}_n = \frac{S_n}{n} = \frac{1}{n} \sum_{k=1}^{K} S_{k,n} \, .$$

Using these definitions we may now define Wald's second identity, a key lemma that we will use throughout the sequel.

Lemma 4.1 (Wald's Second Identity)**.** *Let $(X_t)_{t\in\mathbb{N}}$ be a sequence of $(\mathcal{F}_t; t \geq 1)$-adapted random variables such that $\mathbb{E}[X_t|\mathcal{F}_{t-1}] = \mu$ and $\mathbb{E}\big[(X_t - \mu)^2|\mathcal{F}_{t-1}\big] = \sigma^2$ for any $t \geq 1$. Let $S_n = \sum_{t=1}^n X_t$ be the partial sum of the first n random variables ($n \geq 1$) and $\tau > 0$ be some stopping time w.r.t. $(\mathcal{F}_t; t \geq 1)$.*[3] *Then,*

$$\mathbb{E}\big[(S_\tau - \mu\tau)^2\big] = \sigma^2\mathbb{E}[\tau] \ .$$

Proof. See Theorem 14.3 of (Gut, 2005). □

The majority of the proof for Theorem 4.1, then, is covered by the following lemma.

Lemma 4.2. *Given two separate sequences of random variables $\{X_{i,1}, ..., X_{i,T_i(n)}\}$ and $\{X_{j,1}, ..., X_{j,T_j(n)}\}$ and stopping time $T_i(n)$ and $T_j(n)$ satisfying Lemma 4.1, provided that $\mathbb{E}[X] = \mathbb{E}[Y] = \mu$ it follows that:*

$$\mathbb{E}[(S_{k,n} - T_k(n)\mu)(S_{j,n} - T_j(n)\mu)] = 0$$

(the lemma ultimately follows from the conditional independence of the next observation and the past decisions; the full proof is given in Appendix B.1)

We may now proof the theorem.

proof of Theorem 4.1. Using Lemma 4.2 and the fact that $n = \sum_{k=1}^K T_k(n)$ we may then write the loss for algorithm $\mathcal{A}$ as

$$\begin{aligned}
L_n(\mathcal{A}) &= \frac{1}{n^2}\mathbb{E}\left[\left(\sum_{k=1}^K S_{k,n} - n\mu\right)^2\right] \\
&= \frac{1}{n^2}\mathbb{E}\left[\sum_{k=1}^K (S_{k,n} - T_k(n)\mu)^2 + 2\sum_{k\neq j}(S_{k,n} - T_k(n)\mu)(S_{j,n} - T_j(n)\mu)\right] \\
&= \frac{1}{n^2}\sum_{k=1}^K \mathbb{E}\big[(S_{k,n} - T_k(n)\mu)^2\big] \ .
\end{aligned}$$

From Lemma 4.1 we conclude $\mathbb{E}\big[(S_{k,n} - T_k(n)\mu)^2\big] = \sigma_k^2\mathbb{E}[T_k(n)]$ and thus get

$$L_n(\mathcal{A}) = \frac{1}{n^2}\sum_{k=1}^K \sigma_k^2\mathbb{E}[T_k(n)] \ . \tag{4.12}$$

Using the definition of the normalized MSE-regret and that $\min_{1\leq k\leq K} L_{k,n} = \sigma_{k^*}^2/n$,

$$R_n(\mathcal{A}) = n^2\left(L_n(\mathcal{A}) - \frac{\sigma_{k^*}^2}{n}\right) = \sum_{k=1}^K \mathbb{E}[T_k(n)](\sigma_k^2 - \sigma_{k^*}^2),$$

which concludes the proof. □

[3] τ is called a stopping time w.r.t. $(\mathcal{F}_t; t \geq 1)$ if $\{\tau \leq t\} \in \mathcal{F}_t$ for all $t \geq 1$.

The tight connection between sequential estimation and bandit problems revealed by (4.11) allows one to reduce sequential estimation to the design of bandit strategies and vice versa. Moreover, upper bounds on the regret for any bandit strategy transfer immediately (and vice versa) as formalized by the following theorem.

Theorem 4.2 (Reduction)**.** *Let Assumption 4.1 hold for Ψ. Define a corresponding bandit problem (ν_k) by assigning ν_k as the distribution of $-X_{k,1}^2$. Given an arbitrary allocation strategy $\mathcal{A}$, let $\mathrm{Bandit}(\mathcal{A})$ be the bandit strategy that consults $\mathcal{A}$ to select the next arm after obtaining reward Y_t (assumed nonpositive), based on feeding observations $(-Y_t)^{1/2}$ to $\mathcal{A}$ and copying $\mathcal{A}$'s choices. Then, the bandit-regret of $\mathrm{Bandit}(\mathcal{A})$ in bandit problem (ν_k) is the same as the MSE-regret of $\mathcal{A}$ in estimation problem Ψ. Conversely, given an arbitrary bandit strategy $\mathcal{B}$, let $\mathrm{MC}(\mathcal{B})$ be the allocation strategy that consults $\mathcal{B}$ to select the next sampler after observing $-Y_t^2$, based on feeding rewards Y_t to $\mathcal{B}$ and copying $\mathcal{B}$'s choices. Then the MSE-regret of $\mathrm{MC}(\mathcal{B})$ in estimation problem Ψ is the same as the bandit-regret of $\mathcal{B}$ in bandit problem (ν_k) (where $\mathrm{MC}(\mathcal{B})$ uses the average of observations as its estimate).*

Proof. The result follows from Theorem 4.1 since $\sigma_k^2 = \mathbb{E}[X_{k,1}^2] - \mu^2$ and $\sigma_{k^*}^2 = \mathbb{E}[X_{k^*,1}^2] - \mu^2$ where k^* is the lowest variance sampler, hence $\sigma_k^2 - \sigma_{k^*}^2 = \mathbb{E}[X_{k,1}^2] - \min_{1\le k'\le K} \mathbb{E}[X_{k',1}^2]$. Furthermore, the bandit problem (ν_k) ensures the regret of a procedure that chooses arm k $T_k(n)$ times is $\sum_{k=1}^K \mathbb{E}[T_k(n)]\Delta_k$, where $\Delta_k = \max_{1\le k'\le K} \mathbb{E}[-X_{k',1}^2] - \mathbb{E}[-X_{k,1}^2] = \sigma_k^2 - \sigma_{k^*}^2$. □

More specifically, using Theorem 4.2 we can establish bounds on the MSE-regret for the algorithms mentioned in Section 4.1.

Corollary 4.1 (MSE-Regret Upper Bounds)**.** *As a corollary to Theorem 4.2, if we let Assumption 4.1 hold for $\Psi = (\psi_k)$ where (for simplicity) and assume each ψ_k is supported on $[0, 1]$. Then, after n rounds, $\mathrm{MC}(\mathcal{B})$ achieves the MSE-Regret bound of: (4.4) when using $\mathcal{B} =$ UCB1; (4.6) when using $\mathcal{B} =$ UCB-V with $c_\zeta = 10$; (4.8) when using $\mathcal{B} =$ UCB-KL; and (4.9) when using $\mathcal{B} =$ TS.*[4]

Additionally, due to Theorem 4.2, one can also obtain bounds on the minimax MSE-regret by exploiting the lower bound for bandits in (Auer et al., 2002b). In particular, the UCB-based bandit algorithms above can all be shown to achieve the minimax rate $O(\sqrt{mn})$

[4] Note that to apply the regret bounds from Section 4.1, one has to feed the bandit algorithms with $1 - Y_t^2$ instead of $-Y_t^2$ in Theorem 2. This modification has no effect on the regret.

up to logarithmic factors, immediately implying that the minimax MSE-regret of $\mathrm{MC}(\mathcal{B})$ for $\mathcal{B} \in \{\text{UCB1}, \text{UCB-V}, \text{KL-UCB}\}$ is of order $L_n(\mathrm{MC}(\mathcal{B})) - L_n^* = \tilde{O}(K^{1/2} n^{-(1+\frac{1}{2})})$.[5]

Additionally, this theorem allows us to derive lower bounds for the Monte Carlo setting. First, let $\mathbb{V}(\psi)$ denote the variance of $X \sim \psi$ and let $\mathbb{V}^*(\Psi) = \min_{1 \le k \le K} \mathbb{V}(\psi_k)$. For a family $\mathcal{F}$ of distributions over the reals, let $D_{\inf}(\psi, v, \mathcal{F}) = \inf_{\psi' \in \mathcal{F}: \mathbb{V}(\psi') < v} D(\psi, \psi')$, where $D(\psi, \phi) = \int \log \frac{d\psi}{d\phi}(x)\, d\psi(x)$, if the Radon-Nikodym derivative $d\psi/d\phi$ exists, and ∞ otherwise. Note that $D_{\inf}(\psi, v, \mathcal{F})$ measures how distinguishable ψ is from distributions in $\mathcal{F}$ having smaller variance than v. Further, we let $R_n(\mathcal{A}, \Psi)$ denote the regret of $\mathcal{A}$ on the estimation problem specified using the distributions Ψ.

Theorem 4.3 (MSE-Regret Lower Bound)**.** *Let $\mathcal{F}$ be the set of distributions supported on $[0, 1]$ and assume that $\mathcal{A}$ allocates a subpolynomial fraction to suboptimal samplers for any $\Psi \in \mathcal{F}^K$: i.e., $\mathbb{E}_\Psi[T_k(n)] = O(n^a)$ for all $a > 0$ and k such that $\mathbb{V}(\psi_k) > \mathbb{V}^*(\Psi)$. Then, for any $\Psi \in \mathcal{F}$ where not all variances are equal and $0 < D_{\inf}(\psi_k, \mathbb{V}^*(\Psi), \mathcal{F}) < \infty$ holds whenever $\mathbb{V}(\psi_k) > \mathbb{V}^*(\Psi)$, we have*

$$\liminf_{n \to \infty} \frac{R_n(\mathcal{A}, \Psi)}{\log n} \ge \sum_{k: \mathbb{V}(\psi_k) > \mathbb{V}^*(\Psi)} \frac{\mathbb{V}(\psi_k) - \mathbb{V}^*(\Psi)}{D_{\inf}(\psi_k, \mathbb{V}^*(\Psi), \mathcal{F})}.$$

Proof. The result follows from Theorem 4.2 and (Burnetas and Katehakis, 1996, Proposition 1). □

In summary, these results establish the first finite-time upper bounds for general adaptive Monte Carlo algorithm which are a welcome improvement to the traditional asymptotic analysis in this space. Additionally, the established matching (asymptotic) lower bounds, which concretely establish the optimality of the given allocation algorithms, are a new addition to this space and are clearly much sharper than traditional lower bound results such as Cramer-Rao. There is, however, one important caveat to these lower bounds which should be made clear. Specifically, the general setup implicitly enforces the fact that we are not able to "look inside" the sampler or the target integrand to gain any more insight into the problem, this prevents techniques like adaptive importance sampling given in Section 2.2 from being considered. Additionally, the setup enforces a particular form of estimator, namely the unweighted empirical average of all the samples. This constraint prevents many other forms of estimation such as the class of linear unbiased estimators (weighted means) or even so-called super-efficient estimators which are known to exist for these problems (Lehmann and Casella, 1998).

[5] $\tilde{O}$ denotes the order up to logarithmic factors. To remove such factors one can exploit MOSS (Audibert and Bubeck, 2010).

4.4 Implementational Considerations

As mentioned, implicit in our regret definition is the assumption that the sequential allocation algorithm takes negligible time relative to a base sampler call. While this assumption is likely to hold for the simpler UCB1 and UCB-V algorithms the KL-UCB and Thompson sampling implementations both require non-trivial computation to make a decision. Specifically, the KL-UCB algorithm requires K binary search operations to solve the optimization given in Eq. (4.7) and the Thompson sampling algorithm must draw a sample from K unique beta distributions per step.

In light of these concerns one possible optimization is to modify the bandit algorithms so that they only consider switching from the arm chosen the last time if the number of samples from the chosen arm increased $(1 + c)$-fold, where $c > 0$ is a tuning parameter (Abbasi-Yadkori et al., 2011). This has the effect that the bandit arm selection is called only $O(\log(n))$ times for n rounds, making the cost of the selection indeed negligible to the cost of sampling as n gets large. The effect of using a larger value of c is to decrease the overhead of the sequential procedure (a benefit), while the regret increases by a factor of $(1 + c)$. In practice, a popular choice is $c \in (0.1, 0.2)$.

Also, the algorithms and bounds in Section 4.1 all assume the base samplers produce samples in the range $[0, 1]$ which may not be the case for arbitrary Monte Carlo samplers. Naturally, these approaches may be applied by simply rescaling the sampled values, that is supposing $X_{k,t} \in [a_k, b_k]$ where $a_k < b_k$ are *a priori* known, we define the rescaled values $\tilde{X}_{k,t} = (X_{k,t} - \min_k a_k)/(\max_k b_k - \min_k a_k)$. We may then feed $1 - \tilde{X}^2_{k,t}$ to the bandit algorithms which will not alter the regret beyond constant factors.

However, as these algorithms are designed to prepare for the "worst-case" they are sensitive to the overestimation of the range, this mapping would therefore lead to an unnecessary deterioration of the performance. A better option is to scale each variable separately. In this case the upper-confidence bound based algorithms must be modified by scaling each of the rewards with respect to its own range and then the bounds needs to be scaled back to the original range. Thus, the method that computes the reward upper bounds must be fed with $\frac{(b_k - a_{\min})^2 - (X_{k,l} - a_{\min})^2}{(b_k - a_{\min})^2} \in [0, 1]$, where $a_{\min} = \min_{1 \le k \le K} a_k$. Then, if the method returns an upper bound $B_{k,l}$, the bound to be used for selecting the best arm (which should be an upper bound for $-\mathbb{E}\left[(X_{k,t} - a_{\min})^2\right]$) should be $B'_{k,t} = (b_k - a_{\min})^2 (B_{k,t} - 1)$. Here we exploit the property $-\mathbb{E}\left[(X_{k,t} - a_{\min})^2\right] = -\mathbb{E}\left[X^2_{k,t}\right] + c$ where the constant $c = 2\mu a_{\min} - a^2_{\min}$ (which is neither known, nor needs to be known or estimated) is

common to all the samplers, hence finding the arm that maximizes $-\mathbb{E}\left[(X_{k,t} - a_{\min})^2\right]$ is equivalent to finding the arm that maximizes $-\mathbb{E}\left[X_{k,t}^2\right]$.

Since the confidence bounds will be looser if the range is larger, we thus see that the algorithm may be sensitive to the ranges $r_k = b_k - a_{\min}$. In particular, the $1/n$-term in the bound used by UCB-V (Eq. (4.5)) will scale with r_k^2 and may dominate the bound for smaller values of n. In fact, UCB-V needs $n \approx r_k^2$ samples before this term becomes negligible, which means that for such samplers, the upper confidence bound of UCB-V will be above 1 for r_k^2 steps. Even if we crop the upper confidence bounds at 1, UCB-V will keep using these arms with a large range r_k defending against the possibility that the sample variance crudely underestimates the true variance. Since the bound of KL-UCB is in the range $[0, 1]$, KL-UCB is less exposed to this problem.

Alternatively, we note that the assumption that the samples belong to a known bounded interval is not strictly necessary (it was not used in the reduction results). In fact, the upper-confidence based bandit algorithms mentioned can also be applied when the payoffs are subgaussian with a known subgaussian coefficient or even if the tail is heavier (Bubeck et al., 2011, 2013).[6] In fact, the weaker assumptions under which the multi-armed bandit problem with finitely many arms has been analyzed assumes only that the $1 + \varepsilon$ moment of the payoff is finite for some known ε with a known moment bound (Bubeck et al., 2013). These strategies must replace the sample means with more robust estimators and also modify the way the upper confidence bounds are calculated. In our setting, the condition on the moment transfers to the assumption that the $2 + \varepsilon$ moment $\mathbb{E}\left[|X_{k,t}|^{2+\varepsilon}\right]$ must be finite with a known upper bound for some ε.

4.5 Experimental Evaluation

The reduction given in this chapter raises a number of questions that are best answered through empirical study. Foremost is the question of which bandit methods are most effective for the adaptive Monte Carlo settings. While there are already a handful of empirical studies present in the literature, a large majority of these have focused on the so-called *Bernoulli bandit* setting; where the payoffs are assumed to be Bernoulli distributed. Interestingly, such single-parameter distributions are not relevant to the adaptive Monte Carlo context as the different arms cannot have identical means yet unique variances. This is an important detail since stochastic bandit algorithms such as KL-UCB and TS are known to

[6] A centered random variable X is subgaussian if $\mathbb{P}(|X| \geq t) \leq c \exp(-t^2/(2\sigma^2))$ for all real t with some $c, \sigma > 0$.

perform considerably better than competing methods on these single-parameter distributions. We would like to examine whether this performance advantage translates to adaptive Monte Carlo estimation or not.

A second question is how stochastic bandit methods compare to existing adaptive Monte Carlo methods in practical settings. To provide some insight into this question, we consider a popular financial option pricing task where one is provided with set of viable sampling distributions (proposals) and tasked with approximating an expectation as efficiently as possible. The performance of various bandit methods in choosing the best proposal is contrasted with that of the d-kernel population Monte Carlo approach described in Section 2.2.1. Recall that this approach uses the cumulative importance weights for each sampling distribution as a heuristic for choosing the next sample. This approach makes for a straightforward comparison since the computational cost of producing a single sample remains fixed for all parameter settings. The same cannot be said for more sophisticated techniques such as AMCS and SMCS where individual samples may have different (parameter-dependent) computational costs.

4.5.1 2-Arm Synthetic Experiments

We first consider the performance of allocation strategies on a simple 2-arm problem. Again, since single-parameter distributions cannot be used, our main interest is in how the various bandit techniques compare across the possible payoff distributions. In particular, we consider a preliminary experimental analysis using three straightforward parametric forms: uniform, truncated normal, and what we will call a *scaled Bernoulli* distributions. The scaled Bernoulli distribution is parameterized by a mean μ and variance σ^2 and defined by the random variable $X = \mu + \sigma Z$, where Z is a Rademacher random variable. Note that for $\sigma^2 = 1/4$ this distribution recovers a $p = 1/2$ Bernoulli distribution. Also, all distributions where parameterized so that they are defined over the same compact set: $(0, 1)$.

We evaluated the four bandit strategies detailed in Section 4.1: UCB1, UCB-V, KL-UCB, and TS, where each algorithm was fed the rewards prescribed by Theorem 4.2, i.e. $-X^2$. For UCB-V we used the same settings as (Audibert et al., 2009), and for TS we used the uniform Beta prior, i.e., $\alpha_0 = 1$ and $\beta_0 = 1$. The regret curves with 99% empirical percentiles are plotted in Fig. 4.1 for the setting where each distribution was set to the same mean ($\mu = 0.5$) and unique variances ($\sigma_1^2 = 0.02, \sigma_2^2 = 0.06$). As an answer to our primary question we observe that, while there are subtle differences in these regret curves, for the

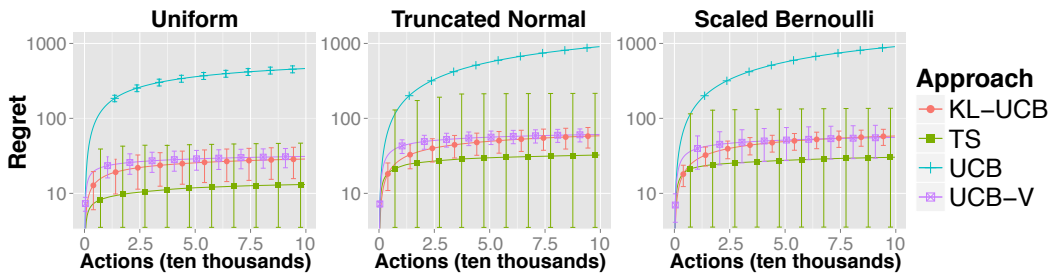

Figure 4.1: The average regret for the different bandit approaches are shown (averaged over 2000 runs) for uniform, truncated normal, and scaled Bernoulli "payout" distributions. Error bars give the 99% empirical percentiles.

most part the shape of the distribution does not have a meaningful effect on the relative performance of these four bandit algorithms. We note also that additional experiments (not shown) using different variance parameterizations yielded similar results.

Considering the fact that the shape of the distribution does not have a strong influence on relative performances we continue with our synthetic evaluations using only the scaled Bernoulli distribution as this distribution permits the maximum range for variance around a mean within a bounded interval. We now turn our attention to understanding how the different approaches perform for different combinations of variances.

Specifically, we consider problem instances where the variance of the first distribution, σ_1^2, is set to values in the range 2^{-i} for $i = 2 : 9$ and the variance of the second arm is set to $\sigma_2^2 = \sigma_1^2 + \Delta$ for $\Delta = 2^{-j}$ and $j = 2 : 9$. Note that in order to ensure the distributions produce samples in the bounded range $(0, 1)$ the variance cannot exceed $1/4$, as a result there are some combinations of i and j that must be omitted, leaving 51 valid combinations. The relative performance of the algorithms are illustrated in Fig. 4.2 for two different time horizons, $n = 5000$ and $n = 10^6$.

From these simulations it is clear that the KL-UCB algorithm is particularly well suited for short time horizons and more so when the variance of the optimal arm is small. However, this advantage diminishes as more samples are accumulated and KL-UCB is eclipsed by either Thompson sampling or UCB-V. The performance with respect to UCB-V is somewhat surprising given both the strong asymptotic guarantees for KL-UCB as well as the simulations results of Cappé et al. (2013) which indicated KL-UCB was far superior. However, the empirical evaluations in this work considered only Bernoulli distributions with a couple different parameterizations and a time horizon of 10^4; evidently this does not tell the full story. A second rather unexpected observation is that the UCB-V algorithm appears to be doing well for rather high variance values on short time horizons, that is, the rightmost area in

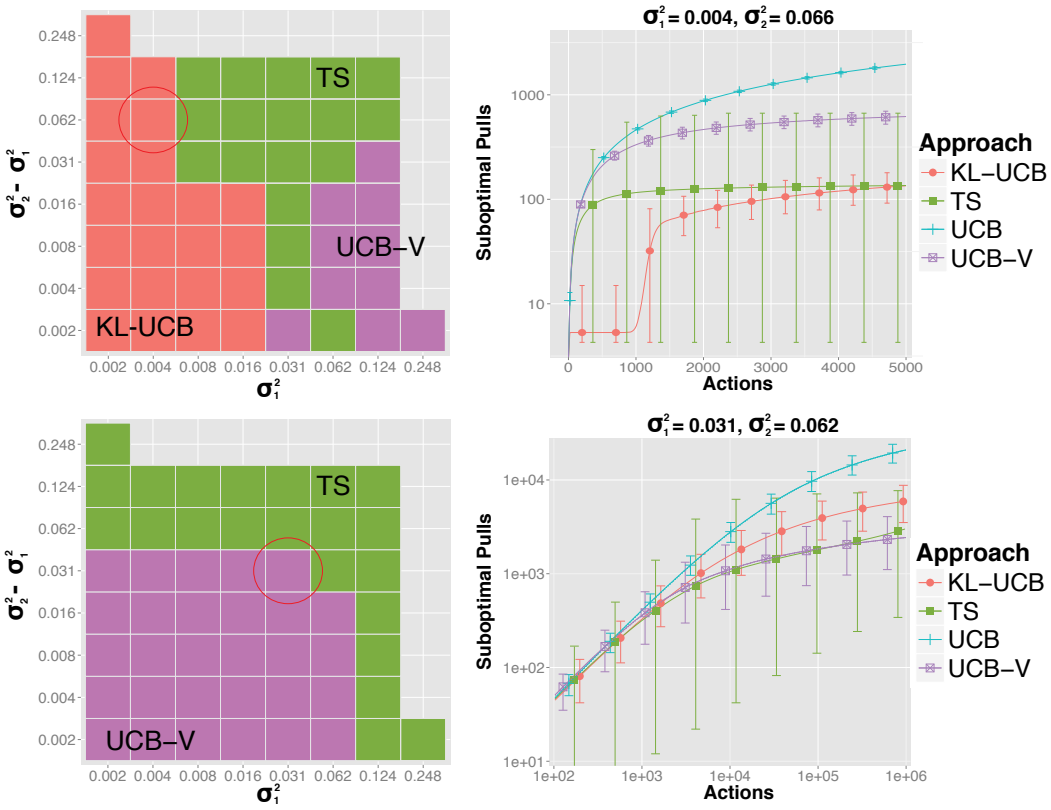

Figure 4.2: **Top left**:Tile plot indicating which approach achieved lowest regret (averaged over 2000 runs) at time step 5000 in the 2-arm scaled Bernoulli setting. X-axis is the variance of the first distribution, and Y-axis is the *additional* variance of the second distribution. **Top right**: Plot illustrating the expected number of suboptimal selections for the highlighted case (dashed red circle in top left plot). Error bars indicate 99% empirical percentiles, Y-axis is log scale. **Bottom left**: Corresponding tile plot taken at time step 10^6. **Bottom right**: Corresponding plot to for time horizon of 10^6, and for a different parameter setting (dashed red circle in bottom left plot). Note that the X and Y axes are log scale.

top-left grid. However, on closer inspection it is revealed that these algorithms (even UCB) had almost exactly the same regret curves for these settings and UCB-V just happened to win by a trivial margin.

As a more general conclusion our results indicate that the Thompson sampling approach is best suited when *either distribution* has high variance for both short and long time horizons. Or, whenever there is a large difference between the arms. On the other hand, for relatively low variance settings, and medium to long time horizons, it is clear that UCB-V is the best choice of algorithm. In regards to the implications for Monte Carlo integration we expect that in most cases Thompson sampling would be the most appropriate choice since there if often a parameter choice that will lead to very high variance samples. In such cases the UCB-V and KL-UCB algorithms do tend to take more time to dismiss these

suboptimal arms.

4.5.2 Option Pricing

We next consider a more practical application of adaptive Monte Carlo estimation to the problem of pricing financial instruments. In particular, following Douc et al. (2007b); Arouna (2004), we consider the problem of pricing *European call options* for an asset that appreciates according to a variable interest rate. Here we assume that the interest rate evolves according to the Cox-Ingersoll-Ross (CIR) model Cox et al. (1985), a popular model in mathematical finance.

In particular, in the CIR model the interest rate at time t, denoted $r(t)$, follows a *square-root diffusion* given by the stochastic differential equation

$$dr(t) = (\eta - \kappa r(t))dt + \sigma\sqrt{r(t)}dW(t),$$

where η, κ and σ are fixed, problem-specific, constants and W is a standard one-dimensional Brownian motion. The payoff of an option for this rate at time t^* (maturity) is given as

$$a\max(r(t^*) - s, 0),$$

where the strike price s and nominee amount a are parameters of the option. The actual quantify of interest, the price of an option at time 0, is given as

$$\mu := \mathbb{E}\left[\exp(-\int_0^{t^*} r(t)dt)a\max(r(t^*) - s, 0)\right].$$

The above integrals may approximated with the Monte Carlo method by first discretizing the interest rate trajectory, $r(t)$, into m points $R^{(1:m)}$ by simulating

$$R^{(t)} = R^{(t-1)} + (\eta - \kappa R^{(t-1)})\frac{t^*}{m} + \sigma\sqrt{R^{(t-1)}\frac{t^*}{m}}\varepsilon_t,$$

where $\varepsilon_t \sim \mathcal{N}(0, 1)$, and $R^{(0)}$ is given. Given a set of n sampled trajectories, $R_{1:n}^{(1:m)}$, we let

$$X_i := \exp\left(-\frac{t^*}{m}\left(\frac{R_i^{(1)} + R_i^{(m)}}{2} + \sum_{t=1}^{m} R_i^{(t)}\right)\right) a\max(r_i^{(m)} - s, 0),$$

and approximate using $\hat{\mu} \approx \frac{1}{n}\sum_{i=1}^{n} X_i$.

An important detail is that the value of the option is exactly zero when the interest rate is below the strike price at maturity. Consequently, when it comes to Monte Carlo simulation, we are less interested in simulating trajectories with lower interest rates. A standard way of exploiting this intuition is an importance sampling variant known as *exponential twisting*

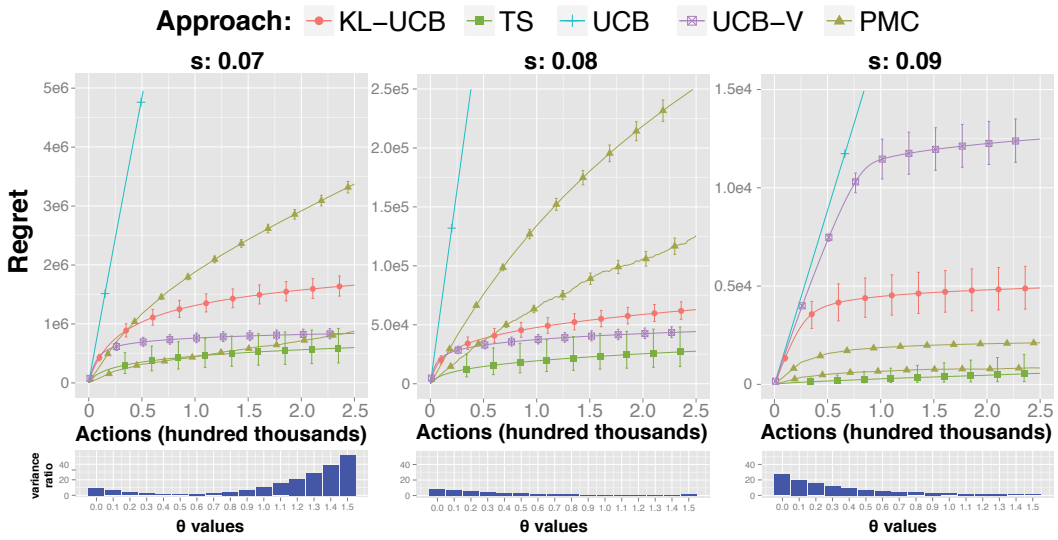

Figure 4.3: **Top:** Regret curves for the different adaptive strategies when estimating the price of a European caplet option at different strike prices (s). Error bars are 99% empirical percentiles. The dashed line is an exception and gives the MSE-Regret of the *Rao-Blackwellized* PMC estimator with error bars delineating 2-standard error. **Bottom:** Bar graphs showing the variance of each sampler (each θ_i-value) normalized by the variance best sampler in the set: $y_i = \mathbb{V}(\hat{\mu}_{\theta_i}) \,/\, \min_j \mathbb{V}(\hat{\mu}_{\theta_j})$.

(see Glasserman (2003)). In this approach, rather than sampling the noise ε_t directly from the target density, $\mathcal{N}(0,1)$, we use a skewed proposal density, $\mathcal{N}(\theta,1)$, defined by a single *drift* parameter, $\theta \in \mathbb{R}$. Deriving the importance weights, $w_i := \exp\left\{\frac{m\theta^2}{2} - \theta \sum_{t=1}^{m} \varepsilon_i^{(t)}\right\}$, then, we arrive at the unbiased approximation $\hat{\mu}_\theta := \frac{1}{n}\sum_{i=1}^{n} w_i X_i$.

Of course, the next question for any practitioner is how to choose the drift parameter θ. In order to automate this choice we consider strategies which select the proposal using a discretized set of parameters, specifically $\theta_j = j/10$ for $j = 1:15$. In addition to evaluating the multi-armed bandit strategies detailed in this chapter we also consider the simple population Monte Carlo (PMC) approach suggested by Douc et al. (2007b). Recall, as detailed in Section 2.2.1, the PMC approach selects proposal densities according to the cumulative importance weights for each proposal. We experimented with option prices under the same parameter settings as (Douc et al., 2007b), namely, $\nu = 0.016$, $\kappa = 0.2$, $R_0 = 0.08$, $t^* = 1$, $a = 1000$, $\sigma = 0.02$, and $n = 100$, for different strike prices $s = \{0.07, 0.08, 0.09\}$. However, we consider a wider set of proposals given by $\theta_k = k/10$ for $k \in \{0, 1, ..., 15\}$. For the bandit methods we used the same configuration as we did in Section 4.5.1 and for PMC we used a population size of $n = 25000$ and $m = 10$ sampling stages. The results, averaged over 10000 simulations, are given in Figure 4.3.

These results indicate Thompson sampling approach is significantly more effective than

the other bandit approaches for this task. Indeed it was the only method that was able to consistently outperform the PMC approach. The performance of the PMC approach is summarized by two curves on each plot; the solid line showing the regret (Eq. (4.10)) of the PMC estimator if it were to be used as a bandit approach, where arms are selected using the resampling heuristic. Alternatively, the dashed line shows the regret of what the authors call the "Rao-Blackwellized" estimator which calculates the importance weights using mixture density over all θ (see Eq. (2.5)). Note that this latter estimator is exploiting domain-specific knowledge that the bandit approaches do not, which essentially allows it to choose actions from a much larger (continuous) space. Again, it is interesting that the allocations produced by Thompson sampling where superior to the extent that they were able to overcome the advantages of PMC.

Despite this strong showing for the bandit methods, PMC remains surprisingly competitive at this task, doing uniformly better than UCB, and taking a close second for the $s = 0.09$ case. From the variance spectrum shown in the bottom bar graphs it seems that the effectiveness of the PMC approach can be at least partly attributed to its ability to quickly dismiss the highly sub-optimal samplers. This being said, one should be cautious of empirical findings of this sort – especially in the absence of theoretical results – since they can indicate the algorithm is over-exploitive, that is, prone to (unluckily) dismissing the optimal sampler too soon. Since this situation results in large regret but occurs very rarely it can be difficult to demonstrate empirically.

4.6 Discussion

In this chapter we have described a general formulation for adaptive Monte Carlo algorithms and subsequently demonstrated that this formulation reduces to the classic stochastic multi-armed bandit formulation. As direct result of this reduction, we immediately have at our disposal a myriad of adaptive approaches many of which enjoy strong finite-time performance guarantees. Further, this reduction permits the application of existing hardness results in the form of asymptotic lower bounds which establishes the optimality of these approaches. From empirical study of the various bandit approaches we have uncovered evidence that each of the tested bandit methods (with the exception of UCB) has its own merits. In particular, KL-UCB is most suitable for short time horizons, UCB-V for low variance settings, and Thompson sampling for high variance setting or setting with highly suboptimal arms. Lastly, our empirical evaluations on the financial pricing example demon-

strates that the bandit approaches, Thompson sampling in particular, are competitive with state of the art adaptive Monte Carlo approaches (PMC) despite the fact that they require far less *a priori* domain knowledge.

Going forward, perhaps the most critical limitation of this bandit formulation is that it does not support adaptation between more sophisticated Monte Carlo methods such as sequential Monte Carlo sampling and antithetic Markov chain sampling. This is due to the assumption that the computational cost of drawing samples from each base sampler (parameterization) is both deterministic and constant amongst all samplers. In the next chapter we address this limitation directly and extend this framework to the stochastic *non-uniform* cost setting.

Another interesting avenue for future research, which we unfortunately do not explore in more detail in this thesis, surrounds settings where the agent may take actions on a continuous range. This general *stochastic optimization* setting houses a number of powerful approaches with some bearing a close resemblance to the methods we considered thus far such as Bayesian optimization routines (Lizzote, 2008) or confidence bound based approaches (Bubeck et al., 2011). In particular, in the adaptive importance sampling example we considered in Section 4.5.2 one may consider actions defined on the 16-dimensional simplex defining mixture distributions (like PMC does) as opposed to only the corners. In fact, in this example the objective function is even continuously differentiable and convex and, as a result, standard local stochastic optimization approaches might perform quite well.

Additionally, in both the discrete and continuous action settings there exists the long-standing challenging ensuring all possible actions correspond to unbiased samplers; Or, said differently, that all proposal densities have a support that overlaps that of the target density. As mentioned in Section 2.2, the standard approach to ensuring full support is *defensive sampling*, where one enforces sampling from a heavy tailed distribution with fixed probability. In many ways this approach is akin the ε-greedy bandit approach and shares many of the main challenges. In particular, this approach will result in slow initial exploration of the integrand as well as unnecessary exploration in later stages unless the rate of *defensive* sampling is carefully managed. One potentially robust approach to managing this exploration might be borrowed from the work on *partial monitoring*; a multi-armed bandit variation where the rewards (variances) for selected actions are not necessarily observed when taking that action. As a result, in this setting, it is often necessary to execute a second (exploratory) action in order to gather the information necessary to evaluate the first. In a way this is analogous to sampling from a heavy tailed distribution as these samples can

actually be used to construct an unbiased estimate for the variance *any* proposal density.

Chapter 5

Adaptive Monte Carlo with Non-Uniform Costs

"Don't think – use the computer."

– George Dyke

In this chapter we extend the *learning to combine Monte Carlo estimators* framework introduced in Chapter 4 to the setting when the base samplers can take different (possible stochastic) amounts of time to generate observations. Our main aim in formulating such an extension is to provide support for more sophisticated Monte Carlo sampling approaches, such as AMCS or SMCS. However, as an added bonus, we note that this general framework is also applicable to numerous other sampling applications. For instance, tuning a rejection sampler is mainly a question of finding parameters that minimize the computational costs, i.e. rejection rate, as opposed to minimizing variance. Additionally, efficient distributed computation over multiple machines in a cluster or data center has become an increasingly important aspect of modern Monte Carlo implementations. Interestingly, hardware differences or uneven resource competition amongst instances may cause Monte Carlo implementations to perform differently across the system. The adaptive framework presented in this chapter is capable of observing these subtle differences and tuning the sampling approaches accordingly, therefore improving the performance and robustness of the entire system.

Naturally, the introduction of unique and randomized sampler computation times, which we refer to as *non-uniform costs*, requires a different regret formulation than the one used in Chapter 4. The framework that we present here follows the basic intuition that, if an sampler takes more time to produce an observation, it is less useful than another sampler that produces observations with, say, identical variance but in less time. Critically, as with

the variances, we do not assume to know the costs associated with each sampler beforehand and they must therefore be approximated online. In particular, any multi-armed bandit strategies one might hope to deploy in this setting must take into account the uncertainty of the variance estimate of each sampler as well as the estimate of the computational costs. Fortunately, as with the previous chapter, we show how a straightforward transformation of the payoffs (costs and samples) permits the direct application of existing stochastic bandit algorithms to this task. We note that some of the work presented in this chapter can be found in the original paper of Neufeld et al. (2014).

5.1 Non-Uniform Cost Formulation

To develop an appropriate notion of regret for this new setting we first introduce some additional notation. In particular, let $D_{k,m} \in \mathbb{R}^+$ denote the time needed by sampler k to produce its m-th sample, $X_{k,m}$. As before, we let $I_m \in \{1, \dots, k\}$ denote the index of the sampler that our allocation algorithm $\mathcal{A}$ chooses in round m, and let $T_k(m) = \sum_{s=1}^{m} \mathbb{I}\{I_s = k\}$ denote the number of samples drawn from k at round m. Let J_m denote the time after $\mathcal{A}$ observes the m-th sample, $Y_m = X_{I_m, T_{I_m}(m)}$, and cost, $D_m = D_{I_m, T_{I_m}(m)}$; that is, $J_0 = 0$, $J_m = \sum_{s=1}^{m} D_{I_s, T_{I_s}(s)}$. We let $N(t) = \sum_{s=0}^{\infty} \mathbb{I}\{J_s \leq t\}$ denote the round number at time $t \in \mathbb{R}^+$ and note that $N(t) > 0$ for all $t \geq 0$.

We assume that $\mathcal{A}$ "receives" the m-th sample immediately at the start of round m and instantaneously updates it estimate. That is, we let $\hat{\mu}(t) = \sum_{i=1}^{N(t)} Y_i$ denote the estimate available at time t. Thus round m starts at time J_{m-1}, with $\mathcal{A}$ choosing an sampler and receiving an observation, and finishes at time J_m. The fact that the algorithm instantaneously observes a new sample after selecting a sampler is a small stretch since, in practice, a Monte Carlo sampler will not actually return a result until it has finished execution. However, this assumption greatly simplifies the following analysis and since the effects of this discrepancy are negligible at any sensible time horizon (where $D_k << t$). The MSE of the estimator used by $\mathcal{A}$ at time $t \geq 0$ is

$$L(\mathcal{A}, t) = \mathbb{E}\left[(\hat{\mu}(t) - \mu)^2\right] .$$

By comparison, the estimate given by a single sampler k at time t is $\hat{\mu}_k(t) = \frac{\sum_{m=1}^{N_k(t)} X_{k,m}}{N_k(t)}$, where $N_k(t) = 1$ on $[0, D_k)$, $N_k(t) = 2$ on $[D_k, D_k + D_{k,2})$, etc. Thus, at time $t \geq 0$ the MSE of the estimator for sampler k is

$$L_k(t) = \mathbb{E}\left[(\hat{\mu}_k(t) - \mu)^2\right] .$$

Given these definitions, it is natural to define the regret as

$$R(\mathcal{A}, t) = t^2 \Big(L(\mathcal{A}, t) - \min_{1 \le k \le K} L_k(t) \Big) . \tag{5.1}$$

As before, the t^2 scaling is chosen so that, under the condition that $L_k(t) \propto 1/t$, a sublinear regret implies that $\mathcal{A}$ is "learning". Note that this definition generalizes the definition in Chapter 4, that is, if $D_{k,m} = 1 \, \forall k, m$, then $R_n(\mathcal{A}) = R(\mathcal{A}, n)$. In what follows we will make use of the following assumption.

Assumption 5.1. *For each k, $(X_{k,m}, D_{k,m})_{(m=1,2,\dots)}$ is an i.i.d. sequence such that $\mathbb{E}[X_k] = \mu$, $\sigma_k^2 := \mathbb{V}(X_k) < \infty$, $D_k > 0$ and $\delta_k := \mathbb{E}[D_k] < \infty$. Furthermore, we assume that the sequences for different k are independent of each other.*

Note that Assumption 5.1 allows for the case where $(X_{k,m}, D_{k,m})$ are dependent in which case $\hat{\mu}_k(t)$ may be a biased estimate of μ. However, if $(X_{k,m})_m$ and $(D_{k,m})_m$ are independent and $\mathbb{P}(N_k(t) > 1) = 1$ then $\hat{\mu}_k(t)$ is unbiased. Indeed, in such a case, $(N_k(t))_t$ is independent of the partial sums $\{\sum_{s=1}^n X_{k,s}\}_{1 \le k \le m)}$, hence $\mathbb{E}[\hat{\mu}_k(t)] = \mathbb{E}\Big[\frac{\sum_{s=1}^{N_k(t)} X_{k,s}}{N_k(t)}\Big] = \sum_{n=1}^{\infty} \mathbb{P}(N_k(t) = n) \, \mathbb{E}\Big[\frac{\sum_{s=1}^n X_{k,s}}{n} \Big| N_k(t) = n\Big] = \mu$. Unfortunately, the same cannot be said of the estimator $\hat{\mu}(t)$ which can be shown to be biased through a simple counter example. In particular, if we consider the 2-arm setting with deterministic costs $D_1 = 1$ and $D_2 = t + 1$ we can see that arm 2 can be chosen at any time to effectively stop the sampling at will. In this case the problem reduces to the classic Chow-Robbins game (Chow and Robbins, 1963), a setting for which bias-maximizing polices are known to exist.

One complication with the regret formulation in Eq. (5.1) is that it is not immediately obvious how to solve for $k^* := \arg\min_{1 \le k \le K} L_k(t)$; indeed k^* may be different for each setting of t. In order to introduce some consistency we consider this optimization in the case where t is sufficiently large. In particular, it follows from the *elementary renewal theorem* for renewal reward processes (Cox et al., 1962) that $L_k(t) \sim \frac{\sigma_k^2/\delta_k}{t(1/\delta_k)^2} = \frac{\sigma_k^2 \delta_k}{t}$, where $f(t) \sim g(t)$ means $\lim_{t \to \infty} f(t)/g(t) = 1$. Intuitively, sampler k will produce approximately t/δ_k independent observations during $[0, t)$, hence the variance of $\hat{\mu}_k(t)$ is approximately $\sigma_k^2/(t/\delta_k)$. As a result, we conclude that for large enough t any allocation strategy $\mathcal{A}$ should draw most of its observations from k satisfying $\delta_k \sigma_k^2 = \delta_{k^*} \sigma_{k^*}^2$, where $k^* := \arg\min_{1 \le k \le K} \delta_k \sigma_k^2$.

As before, we would like a straightforward way to transform the samples and costs so as to facilitate the application of existing sequential allocation algorithms. First, we observe that it is no longer sufficient to use the negative second moment $-X_k^2$ in place of

variance since our objective $\delta_k \sigma_k^2 = \delta_k(\mathbb{E}\left[X_k^2\right] - \mu^2)$ involves the unknown expectation μ. Fortunately, this complication can be bypassed by through the observation that

$$\mathbb{E}\left[\frac{1}{2}(X_{k,m} - X_{k,m+1})^2\right] = \sigma_k^2,$$

which follows from the independence beween samples (Assumption 5.1). As a result, at any round m we can define the bandit payoff for sampler k using two independent draws:

$$Y_{k,m} = -\tfrac{1}{4}(D_{k,2m} + D_{k,2m+1})(X_{k,2m} - X_{k,2m+1})^2, \tag{5.2}$$

where it follows that $\mathbb{E}[Y_{k,m}] = -\delta_k \sigma_k^2$ provided $D_{k,m}$ and $X_{k,m}$ are independent. In the event that this independence does not hold we can instead use a third sample to estimate the costs:

$$Y_{k,m} = -\tfrac{1}{2} D_{k,3m}(X_{k,3m+1} - X_{k,3m+2})^2, \tag{5.3}$$

which achieves the desired expectation. Our strategy then, is to feed a stochastic bandit algorithm with the sequence of rewards $(Y_1, Y_2, ...)$, until the computation budget has been exhausted; recall we do not assume to know this budget beforehand. In this setup we can conclude from the theoretical properties of these allocation algorithms that we will draw samples from the best sampler exponentially more often than any suboptimal sampler (on expectation). We draw more precise conclusions about the finite time regret of such strategies in the next section.

5.2 Bounding the MSE-Regret

Given that the number of suboptimal action made by any particular stochastic bandit algorithms can often be bounded from above (in expectation or with high probability), our aim here will be to upper bound regret of any allocation strategy as a function of the number of suboptimal actions. The main theorem describing this bound is given below.

Theorem 5.1. *Let Assumption 5.1 hold and assume that X_k are bounded and k^* is unique. Let the estimate of $\mathcal{A}$ at time t be defined by the sample mean $\hat{\mu}(t)$. Additionally, assume that for any $k \neq k^*$, $\mathbb{E}[T_k(N(t))] \leq f(t)$ for some $f : (0, \infty) \to [1, \infty)$ where $f(t) \leq C_f t$ for some $C_f > 0$. Assume further that $\mathbb{P}(D_k > s) \leq C_D s^{-2}$ for all $s > 0$. Then, for any $c < \sqrt{t/(4\delta_{\max})}$ where $\delta_{\max} = \max_k \delta_k$, the regret of $\mathcal{A}$ at time t is bounded by*

$$\begin{aligned} R(\mathcal{A}, t) \leq & \left(3\delta_{k^*} c\sqrt{\delta_{max}} + \delta_{k^*} + C_D\right) \sigma_{k^*}^2 \sqrt{t} + C' f(t) \\ & + C''' t^2 \mathbb{P}\left(N_{k^*}(t) > \mathbb{E}[N_{k^*}(t)] + c\sqrt{\mathbb{E}[N_{k^*}(t)]}\right) \\ & + C'''' t^2 \mathbb{P}\left(N(t) < \mathbb{E}[N(t)] - c\sqrt{\mathbb{E}[N(t)]}\right), \end{aligned} \tag{5.4}$$

for some appropriate constants $C', C''', C'''' > 0$ that depend on the problem parameters δ_k, σ_k^2, the upper bound on $|X_{k,m}|$, and the constants C_f and C_D.

We delay the discussion this bound until after we finish the proof, which follows from a series of three lemmas. However, going forward the reader should be aware of one significant aspect of this bound, in particular, the appearance of a $O(\sqrt{t})$ term that is irrespective of the number of suboptimal pulls. This, as we will see, is a rather unfortunate consequence of stochastic costs and can ultimately be traced back to the first key lemma which we state below. Additionally, the condition that the number of samples $N_k(t)$ concentrates about its mean is not easily enforced for many stochastic allocation algorithms, such as Thompson sampling, and requires carefully tuning exploration rates for standard approaches, such as UCB and UCB-V.

Lemma 5.1. *Let S, N be random variables, where $N > 0$ and let $\mu \in \mathbb{R}$. Let $D = \left(\frac{S}{N} - \mu\right)$, and $\mathbb{E}\left[N^4\right] < \infty$. Let $d \in \mathbb{R}$ be an almost sure upper bound on $|D|$, then, for any $c \geq 0$,*

$$\mathbb{E}\left[D^2\right] \leq \frac{\mathbb{E}\left[(S - N\mu)^2\right]}{\mathbb{E}[N]^2}\left(1 + \frac{2c}{\sqrt{\mathbb{E}[N]}}\right)^2 + d^2\left[\mathbb{P}\left(N < \mathbb{E}[N] - c\sqrt{\mathbb{E}[N]}\right) + \mathbb{I}\{\mathbb{E}[N] < 4c^2\}\right]. \tag{5.5}$$

Additionally, we have the lower bound

$$\mathbb{E}\left[D^2\right] \geq \frac{\mathbb{E}\left[(S - N\mu)^2\right]}{\mathbb{E}[N]^2}\left(1 - \frac{2c}{\sqrt{\mathbb{E}[N]}}\right) - d^2\frac{\sqrt{\mathbb{E}[N^4]}}{\mathbb{E}[N]^2}\mathbb{P}\left(N > \mathbb{E}[N] + c\sqrt{\mathbb{E}[N]}\right). \tag{5.6}$$

(The proof exploits properties relating to N concentrating about its mean; full proof given in Appendix C.1)

This lemma is will be used in bounding the losses $L(\mathcal{A}, t)$ and $L_k(t)$ since either can written as expression with the same form as D. This result is useful when N concentrates about its mean at roughly a $O(\sqrt{t})$ rate, which we generally expect for the sum of a i.i.d. sequence. That is, we expect $\mathbb{P}\left(N < \mathbb{E}[N] - C\sqrt{\mathbb{E}[N]\log(1/\delta)}\right) \leq \delta$ with some numerical constant C. Thus, setting $c = C\sqrt{\log(1/\delta)}$ results in

$$d^2\left[\mathbb{P}\left(N < \mathbb{E}[N] - c\sqrt{\mathbb{E}[N]}\right) + \mathbb{I}\{\mathbb{E}[N] < 4c^2\}\right] \leq d^2\delta + d^2\mathbb{I}\{\mathbb{E}[N] < 4C^2\log(1/\delta)\}.$$

Suppose that $\mathbb{E}[N] = rt$ and choose $\delta = t^{-q}$ with $q > 0$, which ensures that the entire expression Eq. (5.4) is $O(\sqrt{(t)})$. Then, as soon as $rt > 4C^2\sqrt{q\log(t)}$, the last indicator

of (5.5) becomes zero. Further, since $\frac{2c}{\sqrt{\mathbb{E}[N]}} = \sqrt{\frac{2q\log(t)}{rt}}$ converges to zero at a $1/\sqrt{t}$ rate one would therefore choose $q \geq 3/2$ to ensure the d^2 terms converge at the same rate as the leading term.

We now turn our attention toward bounding the main term $\mathbb{E}\left[(S - N\mu)^2\right] / \mathbb{E}[N]^2$. In order to upper bound the regret given Eq. (5.1), we will require an upper bound on $L(\mathcal{A}, t)$ and a lower bound on $L_{k^*}(t)$, we proceed first with this lower bound for which the following lemma will be used.

Lemma 5.2. *Let Assumption 5.1 hold and assume that the random variables $|X_{k,m} - \mu|_{k,m}$ are a.s. bounded by some constant $B > 0$. We consider the case where an individual sampler $k \in \{1, \dots, K\}$ is used up to time t: letting $N_k(t)$ be the smallest integer such that $\sum_{m=1}^{N_k(t)} D_{k,m} \geq t$ and $S_k(t) = \sum_{m=1}^{N_k(t)} X_{k,m}$. Then, for any $t \geq 0$ we have $N_k > 0$, also we define constant $\beta > 0$ such that $\mathbb{P}(D_k \leq \beta) > 0$,*

$$\frac{\mathbb{E}\left[(S_k(t) - N_k(t))\mu)^2\right]}{\mathbb{E}[N_k(t)]^2} \geq \frac{\sigma_k^2 \delta_k}{t + \beta} - \frac{\sigma_k^2 \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}]}{t + \beta} \tag{5.7}$$

and

$$\frac{t}{\delta_k} \leq \mathbb{E}[N_k(t)] \leq \frac{t + \beta}{\delta_k - \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}]}, \tag{5.8}$$

(proof makes use of Wald's second identity (Lemma 4.1); full proof is given in Appendix C.2)

The lemma makes use of a high probability upper bound (β) on the costs which may need to be chosen with some care. If the random variables $(D_{k,m})$ are a.s. bounded by a constant, we can simply choose β to be their common upper bound to cancel the third term of (5.7). Otherwise, we can make use of the assumptions in Theorem 5.1, namely $\mathbb{P}(D_k > s) \leq C_D s^{-2}$. Specifically, we observe

$$\begin{aligned}
\mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}] &= \int_0^\infty \mathbb{P}(D_k \mathbb{I}\{D_k > \beta\} > y)\, dy \\
&= \int_0^\beta \mathbb{P}(D_k \mathbb{I}\{D_k > \beta\} > y)\, dy + \int_\beta^\infty \mathbb{P}(D_k \mathbb{I}\{D_k > \beta\} > y)\, dy \\
&\leq \int_\beta^\infty C_D y^{-2} dy = \frac{C_D}{\beta}.
\end{aligned} \tag{5.9}$$

We may then choose $\beta = \sqrt{t}$ which ensures the overal contribution of the third term in Eq. (5.7) will be negligible and that the r.h.s of this expression is $\Omega(t^{-1})$. We now present a similar lemma which we will be used to bound $L(\mathcal{A}, t)$ from above assuming we have an upper bound on the expected number of suboptimal actions.

Lemma 5.3. *Let Assumption 5.1 hold and assume that the random variables $|X_{k,m} - \mu|_{k,m}$ are a.s. bounded by some constant $B > 0$. We consider the case where $\mathcal{A}$ sequentially chooses samplers up to time t: letting $N(t)$ be the smallest integer such that $\sum_{m=1}^{N(t)} D_{I_m,T_{I_m}(m)} \geq t$ and $S(t) = \sum_{m=1}^{N(t)} Y_m$. Additionally, assume that t is large enough to ensure $k^* = \arg\min_{1\leq k\leq K} L_k(t) = \arg\min_{1\leq k\leq K} \delta_k\sigma_k^2$, and that k^* is unique. Let $f(s) \geq \mathbb{E}[T_k(N(s))]$ for $s \geq 0$ and $1 \leq k \leq K$, Then,*

$$\frac{\mathbb{E}\left[(S(t) - N(t)\mu)^2\right]}{\mathbb{E}[N(t)]^2} \leq \frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{C'f(t)}{t^2} + \frac{C''f(t)^2}{t^3} \tag{5.10}$$

for some constants $C', C'' > 0$ that depend only on the problem parameters δ_k, σ_k^2, Furthermore,

$$\mathbb{E}[N(t)] \geq \frac{t}{\delta_{\max}}\,. \tag{5.11}$$

(The proof for Eq. (5.10) is similar to the proof of Theorem 4.1 and Eq. (5.11) follows in much the same way as Eq. (5.7); the full proof is given in Appendix C.3)

We may now prove the main theorem.

Proof of Theorem 5.1. We can use Eq. (5.6) in conjunction with Lemma 5.2 and Lemma 5.3 to bound the difference between the loss of $\mathcal{A}$ and that of the best arm. Defining $\lambda^*(\beta) := \mathbb{E}[D_{k^*}\mathbb{I}\{D_{k^*} > \beta\}]$ we observe

$$\begin{aligned}
L(\mathcal{A},t) - L_{k^*}(t) &\leq \frac{\mathbb{E}\left[(S(t) - N(t)\mu)^2\right]}{\mathbb{E}[N(t)]^2}\left(1 + \frac{2c}{\sqrt{\mathbb{E}[N(t)]}}\right)^2 \\
&\quad - \frac{\mathbb{E}\left[(S_{k^*}(t) - N_{k^*}(t)\mu)^2\right]}{\mathbb{E}[N_{k^*}(t)]^2}\left(1 - \frac{2c}{\sqrt{\mathbb{E}[N_{k^*}(t)]}}\right) \\
&\leq \left(\frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{C'f(t)}{t^2} + \frac{C''f(t)^2}{t^3}\right)\left(1 + \frac{2c\sqrt{\delta_{\max}}}{\sqrt{t}}\right)^2 \\
&\quad - \left(\frac{\delta_{k^*}\sigma_{k^*}^2}{t+\beta} - \frac{\sigma_{k^*}^2\lambda^*(\beta)}{t}\right)\left(1 - \frac{c\sqrt{\delta_{k^*}}}{\sqrt{t}}\right) \\
&= \left(\frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{C'f(t)}{t^2} + \frac{C''f(t)^2}{t^3}\right)\left(1 + \frac{2c\sqrt{\delta_{\max}}}{\sqrt{t}} + \frac{4c^2\delta_{\max}}{t}\right) \\
&\quad - \left(\frac{\delta_{k^*}\sigma_{k^*}^2}{t+\beta} - \frac{\sigma_{k^*}^2\lambda^*(\beta)}{t}\right)\left(1 - \frac{c\sqrt{\delta_{k^*}}}{\sqrt{t}}\right) \\
&\leq \frac{\beta\,\delta_{k^*}\sigma_{k^*}^2}{t(t+\beta)} + \frac{3\delta_{k^*}\sigma_{k^*}^2 c\sqrt{\delta_{max}}}{t^{3/2}} + \frac{C'f(t)}{t^2} + \frac{C''f(t)^2}{t^3} + \frac{\sigma_{k^*}^2\lambda^*(\beta)}{t} \\
&\leq \frac{\beta\,\delta_{k^*}\sigma_{k^*}^2}{t(t+\beta)} + \frac{3\delta_{k^*}\sigma_{k^*}^2 c\sqrt{\delta_{max}}}{t^{3/2}} + \frac{(C' + C''C_f)f(t)}{t^2} + \frac{\sigma_{k^*}^2\lambda^*(\beta)}{t}\,,
\end{aligned}$$

where we absorbed some lower order terms, $O(\frac{f(t)}{t^{5/2}})$ and $O(\frac{f(t)^2}{t^{7/2}})$, into the constants C' and C''. Recall also that either $\lambda^*(\beta) = 0$ and $\beta = B$ (if $D_{k^*} < B$ a.s.) or $\lambda^*(\beta) \leq \frac{C_D}{\beta}$

and $\beta = \sqrt{t}$. Assuming only latter condition we then have

$$L(\mathcal{A}, t) - L_{k^*}(t) \leq \frac{C}{t^{3/2}} + \frac{C' f(t)}{t^2} \tag{5.12}$$

for $C = 3\delta_{k^*}\sigma_{k^*}^2(c\sqrt{\delta_{max}} + 1/3) + \sigma_{k^*}^2 C_D$ and C' redefined to absorb $C''C_f$.

It remains only to modify the trailing terms given in Lemma 5.1, first we see that the terms in Eq. (5.6) become $C'''t^{-2}\sqrt{\mathbb{E}[N_{k^*}(t)^4]}\mathbb{P}\Big(N_{k^*}(t) > \mathbb{E}[N_{k^*}(t)] + c\sqrt{\mathbb{E}[N_{k^*}(t)]}\Big)$. Since $\mathbb{E}[N(t)] \geq t/\delta_{\max}$, the indicator Eq. (5.5) is bounded as $\mathbb{I}\{\mathbb{E}[N(t)] < 4c^2\} \leq \mathbb{I}\{t < 4\delta_{\max}c^2\}$ which is zero since by assumption $c < \sqrt{t/(4\delta_{max})}$. Adding these terms into Eq. (5.12) and multiplying by t^2 as per the regret definition Eq. (5.1) concludes the proof. □

5.2.1 Discussion

In general the results of Theorem 5.1 are promising and allow us to conclude that our reduction from the non-uniform cost setting to the stochastic bandit setting will result in a $O(\sqrt{t})$ regret up to logarithmic factors (see below). To obtain such a rate, one need only achieve $f(t) = O(\sqrt{t})$, which can be attained by stochastic or even adversarial bandit algorithms (Auer et al., 2002b) receiving rewards with expectation $-\delta_k\sigma_k^2$ and a well-concentrated number of samples. The tail assumptions on the cost distribution ($\mathbb{P}(D_k > s) \leq s^{-2}$) is also not restrictive; for example, if the estimators are rejection samplers, their sampling times will have a geometric distribution that easily satisfies the polynomial tail condition. Lastly, this result is consistent with the known $O(\sqrt{t})$ regret bound for the related *bandits with knapsacks* domain which has a similar (more general) construction (see Badanidiyuru et al. (2013)).

Note, however, that ensuring $\mathbb{E}[T_k(N(t))] \leq f(t)$ can be nontrivial. Typical guarantees for UCB-type algorithms ensure that the expected number of pulls to a suboptimal arm k in n rounds is bounded by a function $g_k(n)$. However, due to their dependence, $\mathbb{E}[T_k(N(t))]$ cannot generally be bounded by $g_k(\mathbb{E}[N(t)])$. Nevertheless, if we have that $D_{k,m} \geq \delta^-$ for some $\delta^- > 0$, then $N(t) \leq t/\delta^-$, hence $f_k(t) = g_k(t/\delta^-)$ can be used. This condition will be satisfied in practice since there will always be some constant overhead to executing a Monte Carlo sampler.

Finally, as mentioned perviously we need to ensure that the trailing terms (5.4) remain small, which follows if $N(t)$ and $N_{k^*}(t)$ concentrate around their means. In general, $\mathbb{P}\Big(N < \mathbb{E}[N] - C\sqrt{\mathbb{E}[N]\log(1/\delta)}\Big) \leq \delta$ for some constant C, therefore $c = C\sqrt{\log(1/\delta)}$ can be chosen to achieve $t^2\mathbb{P}\Big(N < \mathbb{E}[N] - c\sqrt{\mathbb{E}[N]}\Big) \leq t^2\delta$, hence by chosing δ to be

$O(t^{-3/2})$ we achieve $\widetilde{O}(\sqrt{t})$ regret, where $\widetilde{O}$ hides the new logarithmic factors introduced by c. Additionally, in order to ensure concentration the allocation strategy must also select the optimal estimator most of the time. For example, Audibert et al. (2009) show that with default parameters, UCB1 and UCB-V will select suboptimal arms with probability $\Omega(1/n)$, making $t^2\mathbb{P}\left(N < \mathbb{E}[N] - c\sqrt{\mathbb{E}[N]}\right) = \Omega(t)$. However, by increasing the constant 2 in UCB1 and the parameter ζ in UCB-V, it follows from (Audibert et al., 2009) that the chance of using *any* suboptimal arm more than $c\log(t)\sqrt{t}$ times can be made smaller than c/t (where c is some problem-dependent constant). Outside of this small probability event, the optimal arm is used $t - cK\log(t)\sqrt{t}$ times, which is sufficient to show concentration of $N(t)$. In summary, we conclude that $\widetilde{O}(\sqrt{t})$ regret can be achieved in Theorem 5.1 with many allocation algorithms under reasonable assumptions.

5.3 Experimental Evaluation

The theoretical results presented above indicate that extending this bandit-allocation framework to the non-uniform cost setting comes without considerable drawbacks. The practical implications of this are exciting as we now understand how to apply standard algorithms for more complex adaptive Monte Carlo algorithms, such as the SMCS and AMCS methods. There are some questions that remain, in particular, which standard bandit approaches tend to perform the best in practice and, of these approaches, is there any indication that the concentration of the number of arm pulls is important in practice. This latter question is particularly interesting since the Thompson sampling approach, which significantly outperformed the other approaches in the previous chapter, almost certainly violates this condition. In this section we consider adaptive variants of the AMCS method presented in Chapter 3, as well as the SMCS and AIS methods.

For these experiments we use the same bandit methods as Chapter 4: UCB, UCV-V, KL-UCB, and TS. The former two methods use an adjusted constant multiplier on the confidence intervals (as suggested by Audibert et al. (2009)) which ensure that the $N_k(t)$ values concentrate about their means with high probability. The KL-UCB algorithm uses the KL-confidence bound for the variance parameter derived in Appendix C.4. Both the KL-UCB and the TS implementations do not come with explicit guarantees that the number of arm pulls concentrates with high probability.

5.3.1 Adaptive Antithetic Markov Chain Sampling

With the addition of the non-uniform cost accounting we are finally able to construct the algorithm envisioned in the first chapter, an adaptive variant of the antithetic Markov chain (AMCS) method. Specifically, we define each of the K samplers as instantiations of AMCS having different parameter settings and let the bandit allocation approaches adapt as needed. In the following empirical analysis we return to the kidnapped robot task considered in Section 3.5.3 and explore how the various bandit approaches perform when tuning the step-size parameters as well as the acceptance threshold parameters. As in this previous section, the threshold parameter is set by first collecting a small preliminary sample and choosing a fixed threshold that "accepts" a fixed percentage of points. Here the sampler are configured to use a threshold parameter that accepted roughly 0.5%, 1.0%, 2.0%, or 3.0% of the points and a step-size parameter of 0.5cm, 1.0cm, or 1.5cm.[1] The cost for these samplers is defined as the number of function evaluations used to compute a particular sample, this is the same cost used in the evaluations in Section 3.5.3. Critically, because the AMCS approach is designed to focus the computational effort in areas of the function with large variance, these costs are quite strongly correlated with the integrand values for the same sample. As a result, it is important that we use the dependent-cost formulation for bandit rewards, as given in Eq. (5.3).

Recall that in the previous experiments a number of different configurations were used where the position of the robot and the number of laser sensor readings was altered. The relative cost-adjusted variance (RCAV) for each of the different parameterizations of the AMCS approach, and for each problem configuration, are shown in Fig. 5.1. These values are relative to the best performing sampler for that setting, for example, for the scenario with 24 laser sensor readings and robot position #6 (bottom left), a step-size of 0.5 and a acceptance rate of 1.0% performed the best (had the lowest $\sigma_k^2\delta_k$) and therefore has RCAV of 1.0. The sampler with step-size 1.5 and acceptance rate of 3.0%, however, had a cost-adjusted variance that was 2.2x as high; this implies that it this sampler would require 2.2x the computational effort to achieve the same accuracy.

The results illustrated in Fig. 5.1 indicate that there exists no fixed parameter setting that is optimal for all problem scenarios. In fact, there is no fixed setting that retains its optimality as the robot is moved to different positions. This observation underscores the importance of an adaptive approach since even in the event that the practitioner is able to

[1] Step-sizes were 1/10 as large for the 3d dimension (rotation) i.e. 1.0cm corresponds to [1.0cm,1.0cm,0.1cm].

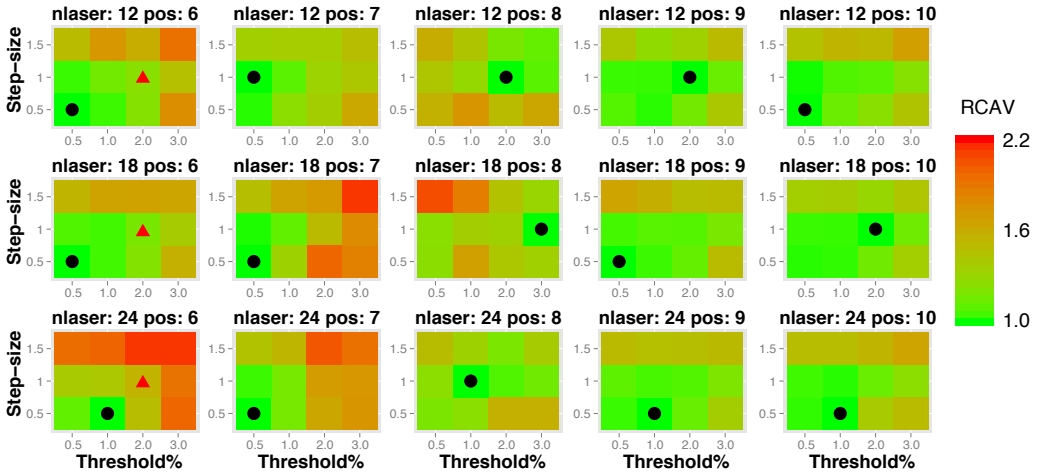

Figure 5.1: Tile plots showing the relative cost-adjusted variance (RCAV), $\sigma^2\delta$, for different parameterizations of AMCS on the various kidnapped robot scenarios used in Section 3.5.3. The measures are relative to the best performing parameterization on that problem which is indicated by the small black circle (here, RCAV = 1.0). The red triangle indicates the constant arm used in the bandit experiments in Fig. 5.2.

painstakingly determine the best parameter setting on average it may still perform worse than an adaptive approach. Additionally, we can see that the performance of the algorithm is quite sensitive to these rather small changes in parameters, since a very small step-size change of 0.5 can easily end up increasing the computational cost by 50%.

The regret curves of the various bandit approaches for this setting, specifically for position #6, are given in Fig. 5.2, these results were gathered over 100 runs. To provide a base comparison the regret of a constant arm – step-size=1.0 and threshold=2.0 – is plotted alongside the adaptive approaches. Note, this arm represents a reasonable choice by any practitioner as performed well across all of the scenarios in Fig. 5.1. The x-axis in these plots is given in terms of the number of samples drawn by the different approaches, to maintain consistency with the experiments in Chapter 3, and a value of $6e + 10$ corresponds to roughly 15hrs of sampling.

As with the previous bandit experiments, it appears that the Thompson sampling approach is best suited for this task as it achieved the best regret in all three settings by a significant margin. Although the approach much more variable performance than the KL-UCB and UCB-V approaches this variance does not appear to be as significant as earlier examples. Interestingly, these simulations do not provide any evidence that high-probability guarantees on $|N_k(t) - \mathbb{E}[N_k(t)]|$ are important for this task, as both TS and KL-UCB performed well. However, any effects resulting from heavy-tailed $N_k(t)$ distributions may not be easily observed from only 100 simulations.

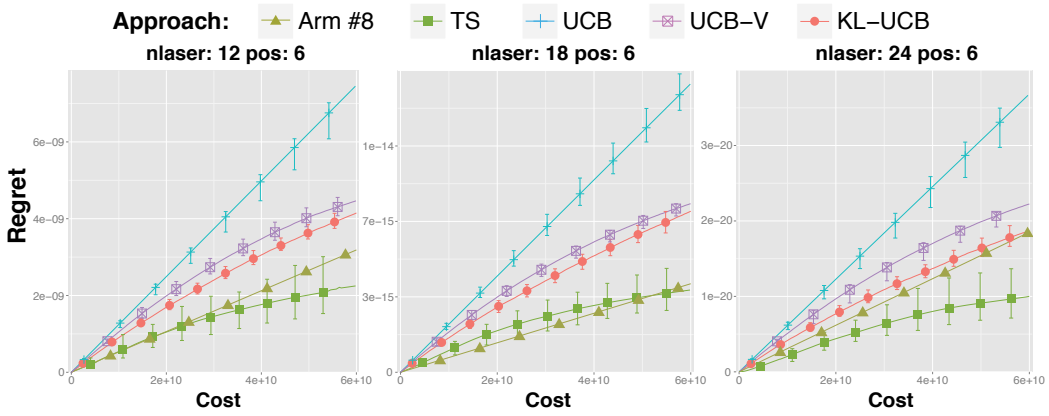

Figure 5.2: Regret curves for the different stochastic MAB algorithms for adaptive-AMCS on the kidnapped robot task. These curves show the performance for a fixed position (#6) as the number of laser sensors are varied (12, 18, 24). For reference we plot the that would be achieved for a fixed parameter choice (marked by the red triangles in Fig. 5.1) which is labelled as Arm #8. Error bars represent 99% empirical density regions.

Ultimately, we can conclude from these experiments that this form of adaptation works quite well for adaptively tuning the parameters for AMCS and can achieve error rates that are unattainable by fixed parameter choices (at least for TS). However, the problem remains difficult for bandit approaches and it is not likely that these approaches would be able to efficiently optimize over extremely large parameter spaces. As a result, a practitioner is likely to extract the most value from this approach by first engaging in some broad prior parameter tuning and leaving the fine-tuning to bandit adaptation.

5.3.2 Tuning Adaptive SMCS

In our review of the adaptive SMCS methods in Section 2.4.2 we describe how the population of samples can also be used to adapt the parameters of the algorithm in an online fashion. Naturally, this approach has unique advantages over the sequential bandit allocation approach proposed in this chapter, and therefore makes for a interesting comparison. However, in this section, rather than compare these approaches directly we evaluate the effectiveness of a far more practical idea, which is to combine these approaches. Specifically, we consider the adaptive approach of Schäfer and Chopin (2013) where the annealing schedule is set automatically by enforcing a minimum effective sample size (ESS) of the particle distribution. While this adaptation is exceedingly good at selecting annealing rates there are a number of parameters for the SMCS approach that are not so easily tuned, such as the number of MCMC transitions in each stage and the population size. Here, we consider using stochastic bandit algorithms for tuning these parameters using independent runs

for these approaches.

Specifically, each arm in the bandit allocation task is defined as a fixed parameter setting for the number of MCMC transitions as well as the number of particles. A single action executes a full simulation of the SMCS approach: using all particles and executing all annealing steps. The weighted value of the final population is then an unbiased sample of the desired quantity and is used as the returned value. The cost for a particular action is determined by CPU time elapsed as reported by the Java VM. This choice for costs is ideal since it is very easily implemented and directly represents likely the costs that the practitioner is interested in. Further, this cost is able to capture almost all aspects affecting the computational performance of the approach such as disk/cache usage to CPU clock speed and even permits effective adaptation across different machines in the same computing cluster.

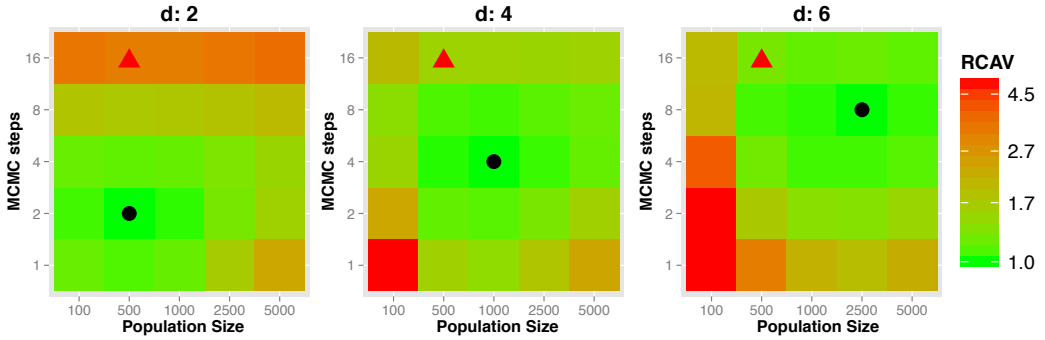

Figure 5.3: Tile plots illustrating the cost-adjusted variance for each parameterization of the adaptive SMCS approach. Values are relative to the best performing parameterization on that problem which is indicated by the small black circle (here, RCAV = 1.0). The red triangle indicates the constant arm used in the bandit experiments in Fig. 5.4.

The integration problem used in these experiments is defined by a simple mixture of 2 un-normalized d-dimensional normal distributions with means $\mu_1 := [1, 1, ..., 1]$ and $\mu_2 := [-1, -1, ..., -1]$ and covariances $0.01I$; that is, $\hat{\pi}(x) \propto \mathcal{N}(x; \mu_1, 0.01I) + \mathcal{N}(x; \mu_2, 0.01I)$. The adaptive SMCS approaches use l slice sampling moves (along a single dimension chosen uniformly at random) and an annealing rate determined by fixing the ESS at a value of $0.8n$, where n is the fixed number of particles. The cross product of the parameters $l \in \{1, 2, 4, 8, 16\}$ and $n \in \{100, 500, 1000, 2500, 5000\}$ defines the settings for each of the 25 bandit arms. The cost-adjusted variance for each of these arms, relative to the best performing arm, are shown in the tile plots in Fig. 5.3. Possibly the most surprising aspect of these plots is that using larger sample sizes is not always better in terms of the tradeoff between CPU usage and variance reduction. This is quite possibly related to the additional computational costs associated with resampling as well as poorer cache performance result-

ing from a larger memory footprint. However, as the dimensionality of the target integrand is increased (becomes more peaked) a larger particle representation becomes more worthwhile and as does using a larger number of MCMC transitions. Additionally, for the bandit experiments, initial experimentation indicated that the costs and sample returns exhibited no measurable correlation, so the more efficient payoff function from Eq. (5.2) was used.

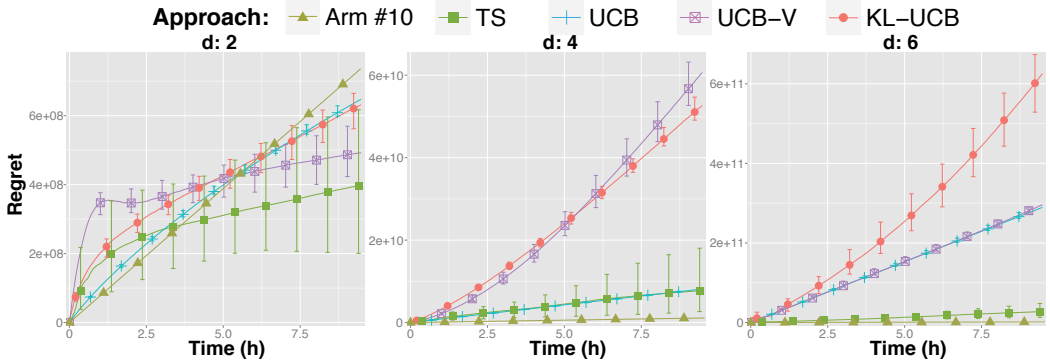

Figure 5.4: Regret curves for the various bandit methods for the adaptive SMCS setting. The regret obtained by pulling only a single arm, with population size 500 and 16 MCMC steps, is labeled Arm #10. X-axis is measured in CPU time as reported by the Java VM and error bars give 99% empirical density regions.

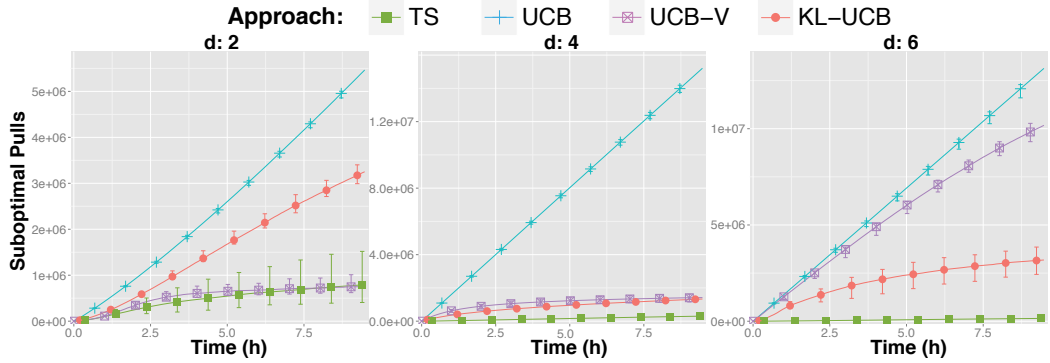

Figure 5.5: Average number of suboptimal arm pulls curves for the various bandit methods for the adaptive SMCS setting. X-axis is measured in CPU time as reported by the Java VM and error bars give 99% empirical density regions. Results are collected over 100 independent simulations.

The regret curves for the various bandit algorithms on this allocation task are shown in Fig. 5.4. These results are highly surprising and entirely unlike the previous bandit experiments. For the easier integration task the bandit methods performed poorly in the first 2-3 hours of sampling before recovering the much more familiar log-shaped regret curve. However, for the higher-dimensional cases, more sampling is required to distinguish

between the arms and as a result, the performance of the bandit approaches was much worse. In fact, the regret curves for KL-UCB and UCB-V appear to be increasing superlinearly, which is somehow a worse rate than simply pulling *any* single arm. More puzzling still, when we look at the number of suboptimal arm pulls for each approach, shown in Fig. 5.5, we see that most of the bandit methods are actually doing a pretty good job of identifying the optimal arm but this does not translate into a lower regret. Indeed, for the d=4 setting we can see that the Thompson sampling approach made far fewer suboptimal arm pulls than UCB but yet almost exactly the same regret curve.

The explanation for this rather counterintuitive effect ultimately comes down to how the samples for each arm are combined into a final estimate. As we mentioned previously, the final estimator takes a simple average of all the collected samples and does not weight them, for example, according to their inverse variances. Said simply, once we have introduced cost into this framework it becomes possible for cheap high-variance samples to effectively "drown out" the expensive low-variance samples. We leave the explanation of this phenomenon until Section 5.3.4 and consider another, more practical, setting in an attempt to ascertain whether this effect is more wide-spread or simply an unfortunate edge-case related to the adaptive SMCS approach.

5.3.3 Adaptively Tuning Annealed Importance Sampling

We next evaluate the effectiveness of the various bandit approaches for adaptively tuning the parameters of an annealed importance sampling (AIS) approach; that is, a SMCS approach with fixed annealing schedule and no resampling. Specifically, the bandit approaches are tasked with allocating computation between 25 unique parameterizations of AIS. Each of these parameterizations is defined by a fixed number of slice sampling MCMC transitions (along a single dimension selected uniformly at random), $l \in \{1, 2, 4, 8, 16\}$, as well as a fixed number of annealing steps, $m \in \{100, 500, 1000, 2500, 5000\}$. In each case, the annealing schedule is defined using the *power of 4* heuristic suggested by (Kuss and Rasmussen, 2005) where the exponent of the is given as $\beta_i = 1 - (i/m)^4$ (see Section 2.4). As with the previous experiment the costs associated with each parameterization are determined by the elapsed CPU time as returned by the Java VM and these costs are assumed to be independent of the sampled returns.

The integration problem used for this evaluation is approximating the normalization constant for a Bayesian logistic regression model having a multivariate Gaussian prior. Specifically, for binary labels $y \in \{0, 1\}$, observations $x \in \mathbb{R}^d$, and latent variables

$\theta \in \mathbb{R}^d$, we let $p(y|x,\theta) = \sigma(\theta^\top x)^y(1-\sigma(\theta^\top x))^{1-y}$, where σ denotes the logit function and let $p(\theta) = \mathcal{N}(\theta; 0, 0.05I)$. The labeled training examples $(x_1, y_1), \ldots, (x_T, y_T)$ are assumed to be generated as follows: first, $\theta_* \sim p(\cdot)$ is sampled, then the sequence $(x_1, y_1), \ldots, (x_T, y_T)$ generated i.i.d. given θ_* and the labels are assumed to satisfy

$$\mathbb{P}(y_t = 1|x_t, \theta_*) = p(y_t|x_t, \theta_*)\,,$$

i.e., $y_t \sim \text{Ber}(\sigma(\theta_*^\top x_t))$. The normalizing constant for the posterior distribution (of which we are ultimately interested in) of θ_* given $x_{1:T}$ and $y_{1:T}$ is given by the integral

$$\zeta_T := \int p_0(\theta) \prod_{t=1}^{T} p(y_t|x_t,\theta) d\theta.$$

The relative performance for the different parameterizations for approximating the normalizing constant for different sized subsets, $T = (5, 10, 25)$ of the 8-dimensional *Pima Indian diabetes* UCI data set (Bache and Lichman, 2013) are shown in Fig. 5.6.

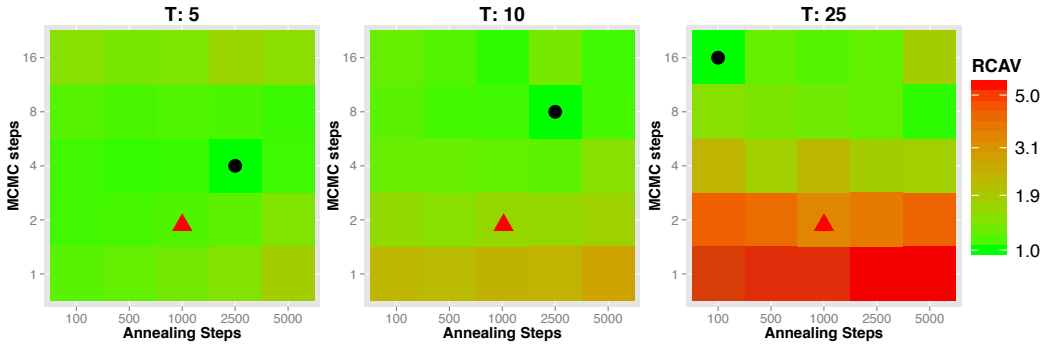

Figure 5.6: Tile plots illustrating the cost-adjusted variance for each parameterization of the AIS approach. Values are relative to the best performing parameterization on that problem which is indicated by the small black circle (here, RCAV = 1.0). The red triangle indicates the constant arm used in the bandit experiments in Fig. 5.7.

The regret curves for the different bandit allocation approaches, on each of the different datasets, are given in Fig. 5.7. Again, we compare the regret for each approach to that of a single arm which performed reasonably well on all datasets. Clearly the problems we observed in the previous SMCS experiments are reflected here as well; in fact these issues are even more pronounced. Specifically, the bandit methods are not able to perform as was as a single fixed arm on any of the three scenarios, even after 24 hours of sampling. Additionally, we see many the regret for the better bandit methods (not UCB) for $T = 10$ and $T = 25$ looks to be increasing at a superlinear rate. This is in despite of the fact that these approaches take considerably fewer suboptimal actions than UCB. Again, this poor

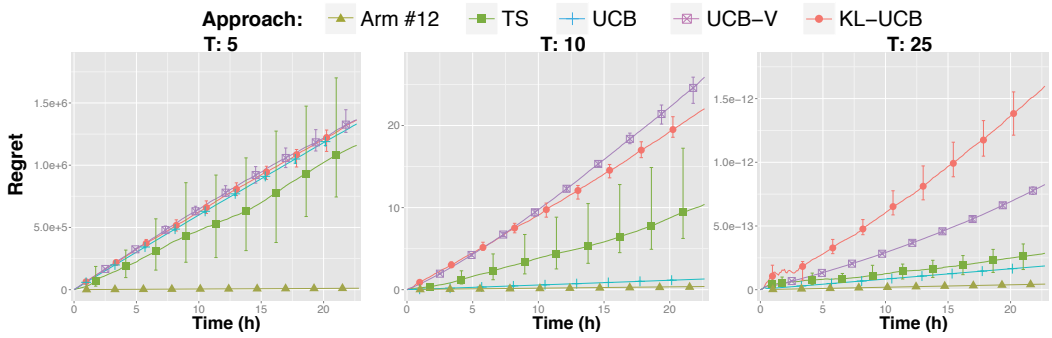

Figure 5.7: Regret curves for the various bandit methods for the AIS setting. The regret obtained by pulling only a single arm, with 100 annealing steps and 2 MCMC steps, is labeled Arm #12. X-axis is measured in CPU time as reported by the Java VM and error bars give 99% empirical density regions. Results are collected over 100 independent simulations.

performance results not from how the computation is allocated but how the samples are combined into a final estimate, as detailed in the next section.

5.3.4 Discussion of Empirical Findings

In order to simplify the regret formulations and subsequent theoretical developments in both of the last two chapters we have assumed that the final estimate is given as a simple arithmetic average of the sampled points irrespective of the distribution for which they were drawn. Specifically, given $N_{1:K}(t)$ samples for each arm and empirical means $\hat{\mu}_{1:K}(t)$ we write the final estimate as:

$$\hat{\mu}(t) = \sum_{k=1}^{K} \lambda_k \hat{\mu}_k(t),$$

where $\lambda_k = \frac{N_k(t)}{N(t)}$, for $N(t) = \sum_{k=1}^{K} N_k(t)$. It straightforward to show that the variance of combined estimate can be reduced further with different λ-weights. In particular, for settings where the samples are drawn deterministically the variance is minimized by weighting the sample means by a value proportional to the inverse their variance (see Theorem 6.1).

The use of uniform weighting can be partially defended by observing that, so long as the bandit allocation draws samples from the optimal arm exponentially more often, the weighting will converge to the optimal weighting asymptotically. However, since in the nonuniform cost setting the best arm is not necessarily the one with the lowest variance, it can take a great deal of time for the uniform weighting to converge to optimal arm, also a bandit approach will not converge to the optimal weighting at all if the optimal arm is not unique.

To demonstrate how the uniform weighting can perform poorly we consider a simple two arm scenario with deterministically chosen sample sizes and constant costs. Specifi-

cally, we define the variance for the arms as $\sigma_1^2 = 1.0$ and $\sigma_2^2 = D$ and the costs as $\delta_1 = 1$ and $\delta_2 = 1/D$, for some positive constant D. Here, it is clear that the two arms are equal in terms of cost-adjusted variance and one might therefore expect the regret for any allocation strategy to be zero. However, consider a deterministic strategy which allocates a fixed proportion of its computational budget (t) to each arm. That is, for some $\gamma \in (0, 1)$ we have $N_1(t) = (1-\gamma)t$ and $N_2(t) = \gamma t D$ (ignoring rounding effects). For this setting we can compute the regret exactly:

$$\begin{aligned} R_t &= t^2 \left(\mathbb{V}\left(\frac{1}{N_1(t) + N_2(t)} \left(\sum_{i=1}^{N_1(t)} X_{1,i} + \sum_{i=1}^{N_2(t)} X_{2,i} \right) \right) - \frac{1}{t} \right) \\ &= t^2 \left(\frac{1}{(N_1(t) + N_2(t))^2} (N_1(t) + N_2(t)D) - \frac{1}{t} \right) \\ &= t \left(\frac{(1-\gamma) + \gamma D^2}{((1-\gamma) + \gamma D)^2} - 1 \right). \end{aligned}$$

The critical observation here is that for any strategy that does not allocate all computation to a single arm, that is for any $0 < \gamma < 1$, this regret is linear in t. Further, as illustrated in Fig. 5.8, as the strategy begins to allocate an increasing amount of computation toward any one sampler the constant multiplier on this linear term may increases rapidly before sharply dropping to zero. This effect explains why the regret in the bandit settings above appears to be growing at a superlinear rate. Additionally, this explains why a suboptimal allocation policy, UCB for instance, which allocates roughly according to $\gamma = 0.5$ may outperform a superior policy which allocates according to, say, $\gamma = 0.98$. Lastly, we can see that as the costs and variances of the arms become more skewed (as D moves away from 1) this effect will become even stronger.

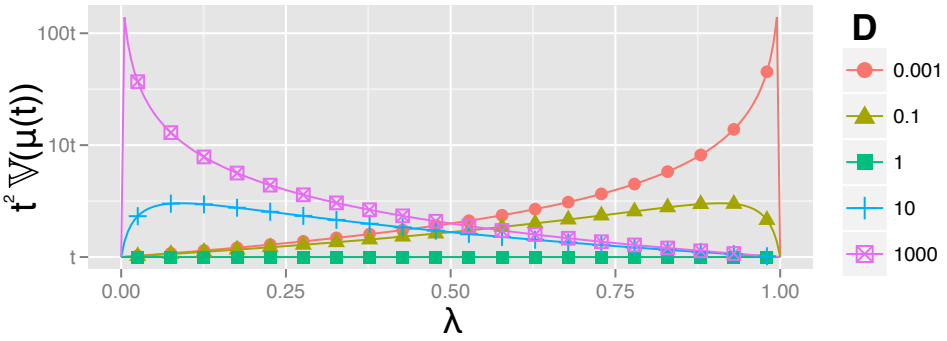

Figure 5.8: Normalized variance (log scale) of the deterministic allocation scheme for different allocation parameter γ and arm 2 variance/cost parameter D.

To answer the question of why the performance for the bandit approaches was consid-

erably better for the AMCS setting than for the SMCS settings, we can look at the costs and variances of the different samplers. Said simply, the SMCS variants all exhibit a similar cost/variance tradeoff, where doubling the amount of computational costs would translate roughly to halving the variance of the sampler plus a small delta. That is to say, the arms were all close in terms of RCAV ($\delta_k \sigma_k^2$) but had vastly different costs and variances, similar to the illustrative example above.

5.4 Discussion

In this chapter we present an extension to the bandit-based adaptive Monte Carlo framework presented in Chapter 4 in which the (potentially stochastic) computational costs associated with each sampler can be accounted for. We formulate a straightforward definition of regret in this scenario and prove a $\widetilde{O}(\sqrt{t})$ upper bound on this regret is attainable with existing sequential allocation approaches. Interestingly, this upper bound does not match the optimal rate for simpler setting (fixed costs) which is known to be $\Omega(\log(t))$. This leaves open the question of whether a tighter upper bound may exist or whether the stochastic cost setting is indeed inherently more difficult than the uniform-cost setting. While we have no specific theoretical arguments, despite considerable effort, we feel that it is most likely that this result is unimprovable. However, the condition that the number of pulls for each arm concentrate around their expectation with high-probability is potentially an artifact of the proof technique, and therefore is an interesting avenue for future research.

In addition to the theoretical findings in this chapter we provide a straightforward way in which the existing stochastic multi-armed bandit approaches can be applied to the non-uniform cost settings. We then evaluate the performance of these approaches by effectively defining an adaptive variant of the antithetic Markov chain sampling approach presented in Chapter 3. Using the same challenging kidnapped robot test setting we show that this bandit-based adaptive algorithm is likely to outperform any fixed parameterization of this method in addition to minimizing the manual tuning work required of the practitioner.

Despite the positive results for adaptive AMCS we find that the adaptive allocation approach may perform poorly in some settings. In particular, we show that the bandit allocation approach is not always able to effectively tune adaptive SMCS or AIS methods in challenging scenarios. This poor performance is determined to not the result of inefficient allocation behaviour but rather a result of procedure for combining these samples into a final estimate. In the next chapter we consider more sophisticated ways to combine a set of

unbiased estimators which is directly applicable to these scenarios and many other practical settings.

Chapter 6

Weighted Estimation of a Common Mean

"An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem."

– John Tukey

In this chapter we consider the task of combining a set of unbiased Monte Carlo estimators for common mean with the goal of minimizing the squared error (MSE) of a final estimate. In contrast to the previous two chapters where we allocated computation to different samplers, in this chapter we assume that all of the samples have already been allocated. The focus therefore remains only on the construction of a final estimate, which we define as a weighted sum of these estimators.

While the construction of a final estimate is a natural extension to sequential allocation tasks, it also arises in classic statistical estimation problems. For example, when aggregating opinion polls that approximate a given statistic, such as the outcome of a presidential election. Typically these polls are conducted by a variety of parties and exhibit a varying degree of accuracy, due to variations in sample size, data collection strategies, or the statistical model used. If we assume that each of the surveys has negligible bias, then the quality of the approximation is a function of the variance of each estimator. For that reason, if estimators with higher variance are given less weight when constructing an estimate, the accuracy of the combined estimate is improved.

In what follows, we show that in cases where the variance of each estimator is known, weighting each individual estimator inversely proportional to its variance recovers both a minimax estimator as well as a minimum variance unbiased estimator. However, as before, we do not assume to know the variances of each estimator beforehand, instead we require

only that they are defined as an empirical average of i.i.d. samples. In considering practically applicable approaches for the unknown variance setting we observe that the simple strategy of weighting the estimators inversely proportional to their sample variances, also known as the *Graybill-Deal* estimator, risks overweighing due to statistical error in sample variance estimates. In order to mitigate these risks, we propose a strategy similar to that used as the multi-armed bandit setting. Specifically, we present an estimator (UCB-W) which uses the upper confidence bound on the sample variance in the weighting formula. We show that this estimator offers significant empirical advantages and permits the construction of straightforward finite-time performance guarantees.

We also show that the proposed UCB-W estimator is even more effective when paired with the adaptive Monte Carlo allocation strategies discussed in the previous chapters. We demonstrate empirically that these techniques offer a far greater reduction in error than the choice of underlying allocation approach (i.e. amongst UCB, TS, etc.). We also show analytically that this estimator can prevent the poor results discussed in Section 5.3.4 where, in the non-uniform cost setting, bandit allocation methods can suffer linear regret. That is, by using the UCB-W estimator one can to retain the original $\widetilde{O}(\sqrt{t})$ bound on regret.

6.1 Weighted Estimator Formulation

Much like the formulation in Chapter 4 we assume that we are presented with a finite set of K unbiased Monte Carlo samplers where each base sampler $k \in \{1, ..., K\}$ produces an i.i.d. sequence of real-valued random variables $\{X_{k,t}\}_{t=1,2,...}$ having the same unknown mean $\mathbb{E}[X_{k,t}] = \mu$ but unique variances $\mathbb{V}(X_{k,t}) = \sigma_k^2 < \infty$. Additionally, in order to facilitate straightforward analysis we will first consider the *deterministic allocation setting* where the number of samples for each estimator are allocated deterministically ahead of time.

More generally, we refer to a K-tuple of distributions $(\nu_1, \ldots, \nu_K)$ over the reals with a common mean μ as an *environment*. The set of all such K-tuples is denoted by $\mathcal{E}(\mu)$. Given the sample sizes $\mathbf{n} := (n_1, \ldots, n_K) \in \mathbb{N}^K$, where $n_k > 0$ and $n := \sum_{k=1}^K n_k$, an *estimator* is a function that maps $\mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_K}$ to the reals.[1] The set of estimators is denoted by $\mathcal{A}_{\mathbf{n}}$. An environment $\nu := (\nu_1, \ldots, \nu_K)$ together with the sample sizes gives rise to the data $X_{\mathbf{n}} := ((X_{k,1}, \ldots, X_{k,n_k}))_{1 \le k \le K}$, where for $k \neq j$, $(X_{k,1}, \ldots, X_{k,n_k})$ and $(X_{j,1}, \ldots, X_{j,n_j})$ are independent of each other and for $1 \le k \le K$, $(X_{k,1}, \ldots, X_{k,n_k})$ is

[1] For simplicity, we do not consider randomizer estimators here.

i.i.d. with common distribution ν_k. In short, we will denote that $X_{\mathbf{n}}$ is generated from ν with sample sizes $\mathbf{n} = (n_1, \ldots, n_K)$ using $X_{\mathbf{n}} \sim \nu^{\mathbf{n}}$. We modify the notation for the loss to indicate its dependence on ν: The loss of estimator $\hat{\mu} \in \mathcal{A}_{\mathbf{n}}$ on environment $\nu \in \mathcal{E}(\mu)$ and sample sizes $\mathbf{n}$ is $L_{\mathbf{n}}(\hat{\mu}; \nu) = \mathbb{E}\left[(\hat{\mu}(X_{\mathbf{n}}) - \mu)^2\right]$ where $X_{\mathbf{n}} \sim \nu^{\mathbf{n}}$.

As with the bandit setting, we consider the loss of an estimator with respect to the "best possible" estimator. In particular, we will consider the following normalized regret $\hat{\mu}^*$, or *excess risk*, formulation

$$R_n(\hat{\mu}; \nu) := n^2 \left(L_n(\hat{\mu}; \nu) - L_n(\hat{\mu}^*; \nu)\right), \tag{6.1}$$

where $\hat{\mu}^*$ is analogous to the notion of the idealized *best arm in hindsight* metric used in the stochastic bandit setting. Here, however there are a number of metrics that might be used to determine the optimality of $\hat{\mu}^*$ fortunately the two most natural notions of optimality: the minimum variance unbiased estimation and minimax estimation, lead to the same estimator (detailed below). We begin by introducing the definitions of minimax and uniform minimum variance unbiased (UMVU) estimators.

Definition 6.1 (Minimax Estimator)**.** *Given a set of environments $\mathcal{E} \subset \cup_{\mu \in \mathbb{R}} \mathcal{E}(\mu)$ and a tuple of sample sizes $\mathbf{n} = (n_1, \ldots, n_K)$, an estimator $\hat{\mu} \in \mathcal{A}_{\mathbf{n}}$ is called minimax w.r.t. $\mathcal{E}$ (or, in short, minimax), if $\sup_{\nu \in \mathcal{E}} L_{\mathbf{n}}(\hat{\mu}; \nu) = \inf_{a \in \mathcal{A}_{\mathbf{n}}} \sup_{\nu \in \mathcal{E}} L_{\mathbf{n}}(a; \nu)$.*

As follows from the definition, a minimax estimator is robust in the sense that its worst-case loss is the best possible.

Next, we consider uniform minimum variance unbiased estimators. First, we say that an estimator $a \in \mathcal{A}_{\mathbf{n}}$ is *unbiased* w.r.t. a set $\mathcal{E}$ of environments if for any $\mu \in \mathbb{R}$, $\nu \in \mathcal{E} \cap \mathcal{E}(\mu)$, $\mathbb{E}[a(X_{\mathbf{n}})] = \mu$ holds where $X_{\mathbf{n}} \sim \nu^{\mathbf{n}}$.

Definition 6.2 (Uniform Minimum Variance Unbiased (UMVU) Estimator)**.** *Given a set of environments $\mathcal{E} \subset \cup_{\mu \in \mathbb{R}} \mathcal{E}(\mu)$ and a tuple of sample sizes $\mathbf{n} = (n_1, \ldots, n_K)$, we say that the estimator $\hat{\mu} \in \mathcal{A}_{\mathbf{n}}$ is a uniform minimum variance unbiased (UMVU) estimator if it is unbiased and for any other unbiased estimator $a \in \mathcal{A}_{\mathbf{n}}$ and for any environment $\nu \in \mathcal{E}$, $\mathbb{V}[\hat{\mu}(X_{\mathbf{n}})] \leq \mathbb{V}[a(X_{\mathbf{n}})]$ where $X_{\mathbf{n}} \sim \nu^{\mathbf{n}}$.*

Note that since the variance of an unbiased estimator is the same as its loss (as we consider quadratic losses), the UMVU estimator achieves the smallest possible expected loss over $\mathcal{E}$ amongst all unbiased estimators.

Let $\sigma^2 = (\sigma_1, \ldots, \sigma_K) \in [0, \infty)^K$. Denote by $\mathcal{E}^{\text{Normal}}_{\sigma^2}$ the set of environments $\nu = (\nu_1, \ldots, \nu_K)$ where each ν_i is normal with variance σ_i^2. Further, denote by $\mathcal{E}_{\sigma^2}$ the set of

environments $\nu = (\nu_1, \ldots, \nu_K)$ where each ν_i has variance σ_i^2. Fix $\mathbf{n} = (n_1, \ldots, n_K)$ and define the estimator $\hat{\mu}^*$ as follows: For $x_{\mathbf{n}} \in \mathbb{R}^{n_1} \times \ldots \mathbb{R}^{n_K}$,

$$\hat{\mu}^*(x_{\mathbf{n}}) := \sum_{k=1}^{K} \lambda_k^* \hat{\mu}_k^*(x_{\mathbf{n}}), \tag{6.2}$$

where $\lambda_k^* := \frac{n_k/\sigma_k^2}{\sum_{j=1}^K n_j/\sigma_j^2}$ and $\hat{\mu}_k^*(x_{\mathbf{n}}) := \frac{1}{n_k} \sum_{t=1}^{n_k} x_{k,t}$.

Theorem 6.1. *The estimator $\hat{\mu}^*$ defined in Eq. (6.2) is* **(a)** *an UMVU estimator w.r.t. $\mathcal{E}_{\sigma^2}^{\text{Normal}}$ and* **(b)** *a minimax estimator w.r.t. any $\mathcal{E}$ such that $\mathcal{E}_{\sigma^2}^{\text{Normal}} \subset \mathcal{E} \subset \mathcal{E}_{\sigma^2}$.*
(The proof follows from standard results given by Lehmann and Casella (1998); full proof is given in Appendix D.1.)

Based on this theorem, we choose the estimator $\hat{\mu}^*$ in the regret definition (6.1) to be the weighted estimator (6.2), giving rise to the explicit regret formulation

$$R_n(\hat{\mu}) := n^2 \left(L_n(\hat{\mu}) - \left(\sum_{k=1}^{K} \frac{n_k}{\sigma_k^2} \right)^{-1} \right), \tag{6.3}$$

where, for brevity, we leave out the notation for the environment ν.

With the understanding that the best estimator weights each sampler inversely proportional to its true variance a natural approach when these variances are unknown is to use the sample variances instead. This choice leads to the classic Graybill-Deal (GD) estimator (Graybill and Deal, 1959) which can be expressed as a convex combination of the individual sample means (as with Eq. (6.2)) as:

$$\hat{\mu}_{GD} := \sum_{k=1}^{K} \lambda_k \hat{\mu}_k \qquad \text{with} \qquad \lambda_k = \frac{n_k/\hat{\sigma}_k^2}{\sum_{j=1}^K n_j/\hat{\sigma}_j^2}, \tag{6.4}$$

where $\hat{\sigma}_k^2 := \frac{1}{n_k - 1} \sum_{t=1}^{n_k} X_{k,t}^2 - \hat{\mu}_k^2$. Here, and in what follows, we will use the shorthand $\hat{\mu}_k \equiv \hat{\mu}_k(X_{\mathbf{n}})$ and $\hat{\sigma}_k^2 \equiv \hat{\sigma}_k^2(X_{\mathbf{n}})$. This estimator was originally suggested for use in the setting where the individual distributions for each arm are normal. The estimator is particularly well suited for this setting since, thanks to Cochran's theorem, we know the sample variances are independent of the sample means and as a result the estimator is unbiased. Additionally, this independence somewhat simplifies the analysis and it can be shown that, for $K = 2$, the estimator is uniformly better than either individual estimator, $\hat{\mu}_1$ or $\hat{\mu}_2$, for any finite $n_1, n_2 \geq 10$ (Graybill and Deal, 1959) (see also Ma et al. 2011 for similar results for $K > 2$). In the statistics literature there are a number of works uncovering various other

properties of the Graybill-Deal estimator (see Mitra and Sinha (2007) for review). However, these analyses do not extend to the non-parametric setting, nor do they give a bound on the regret.

Perhaps this is not surprising, since the estimator can be seen to perform almost arbitrarily poorly in simple examples. For instance, consider the scenario where the X_k are distributed according to a scaled Bernoulli distribution (Section 4.5.1). Here, the probability that the sample variance is exactly zero for any arm k is 2^{-n_k+1}. If we avoid division by zero by defining the estimator to give all weight to $\hat{\mu}_k$ when $\hat{\sigma}_k^2 = 0$, then we can see that for any finite n_k estimator has nonzero probability of assigning all weight to $\hat{\mu}_k$. As a result, we can see that unless all $n_k \to \infty$ as $n \to \infty$ the estimator will suffer quadratic regret. The natural solution to this is to add a small positive constant to the sample variance that decays as n_k grows. We employ this strategy here. In particular we define the upper-confidence bound weighted (UCB-W) estimator:

$$\hat{\mu}_{\text{UCB-W}} := \sum_{k=1}^{K} \lambda_k \hat{\mu}_k \qquad \text{with} \qquad \lambda_k = \frac{n_k/(\hat{\sigma}_k^2 + \Delta_k)}{\sum_{j=1}^{K} n_j/(\hat{\sigma}_j^2 + \Delta_k)}, \tag{6.5}$$

where Δ_k is a confidence bound; we make precise the requirements on Δ_k in the next section. Unlike the Graybill-Deal estimator it is easy to see that this estimator is consistent and asymptotically optimal ($\hat{\mu}_{\text{UCB-W}} \to \hat{\mu}^*$) as $n \to \infty$ regardless of whether any n_k remain finite. In addition to these asymptotic properties we show in the next section that it is possible to construct finite-time upper bounds on the regret of this estimator.

6.2 Bounding MSE-Regret

In this section we provide upper bounds on the regret (Eq. (6.1)) of the UCB-W estimator given in Eq. (6.5). In what follows we assume that the samples $X_{k,t}$ are bounded a.s. so that we may define straightforward confidence intervals. Again, it is possible to relax these assumptions to include sub-Gaussian random variables. In particular, we let $\bar{b}_k = \operatorname{ess\,sup} |X_k|$, it follows from the Hoeffding-Azuma inequality Lemma D.1 and the union bound that, if $\Delta_k(\delta, n_k) = 5\bar{b}_k^2\sqrt{\frac{\log(\delta/(4K))}{2n_k}}$ $0 < \delta < 1$, then $\mathbb{P}(\xi) \geq 1 - \delta$ where

$$\xi := \bigcap_{k=1}^{K} \xi_k$$
$$\xi_k := \left\{ |\hat{\sigma}_k^2 - \sigma_k^2| \leq \Delta(\delta, n_k) \right\}$$

(see Lemma 2 Antos et al. (2010)). Using this definition we can bound the regret for the UCB-W, as formalized in the following theorem.

Theorem 6.2. *Given K sequences of i.i.d. samples $(X_{k,1}, ..., X_{k,n_k})$, $k \in \{1, \dots, K\}$, for deterministically chosen $n_k \geq 2$ where $\operatorname{ess\,sup} |X_k| = \bar{b}_k < \infty$, $\mu = \mathbb{E}[X_k]$, and $\sigma_k^2 < \infty$ the UCB-W estimator given in Eq. (6.5) with $\Delta_k := \Delta_k(\delta, n_k)$ has regret bounded as*

$$R_n(\hat{\mu}_{\text{UCB-W}}) \leq C\sqrt{n} \sum_i \frac{\bar{b}_i \sqrt{\rho_i \log\left(\frac{4K}{\delta}\right)}}{\sigma_i^4} \left(1 + \sum_i \lambda_i^* \frac{\Delta_i}{\sigma_i^2}\right) + 4b^2 \delta n^2 + C' \,,$$

where $b = \max_k \bar{b}_k$, $\rho_k := \frac{n_k}{n}$ and $C, C' > 0$ are constants that depend only on $(\sigma_k)_k$, $(\bar{b}_k)_k$, $(\rho_k)_k$, and K. In particular, one may choose $C = 10\sqrt{2}(\sum_k \frac{\rho_k}{\sigma_k^2})^{-2}$.
(Proof is given in Appendix D.2)

This result shows that the regret will grow at a rate of $\widetilde{O}(\sqrt{n})$, which is in line with our expectations since the sample variances are converging at a rate of $\widetilde{O}(n^{-1/2})$. In general this is a strong result and the rate is considerably better than that achieved by the uniformly weighted estimator, where $\lambda_k = \frac{n_k}{n}$, used in Chapter 4 and Chapter 5. Recall that this estimator will suffer a liner regret in cases where at least two distributions (having different variances) are sampled as a linear function of n.

6.3 Non-Deterministic (Bandit) Formulation

In order to apply the upper confidence weighted estimator to multi-armed bandit settings presented in previous chapters we consider the case where the number of samples for each sampler is stochastic. In particular, we assume that we are provided with K sequences of i.i.d. samples $\{X_{k,1}, ..., X_{k,N_k}\}_{1 \leq k \leq K}$ where the random variables $N_k(t) \geq 2$ denote the number of samples drawn from sampler k at time $t > 0$ in the non-uniform cost setting or $T_k(t)$ in the uniform cost setting (where $t = n$). However, in this setting the actual costs of collecting the individual samples do not enter into the regret, since the samples have already been allocated. As a result, the analysis does not depend on the distribution of costs other than the requirement that $\mathbb{E}[N_k(t)] \leq C_f t$ for some constant C_f, which will be used to bound the regret as a function of t. Additionally, since we will always index by the same value of t we simply drop the explicit indexing notation for N that is, we let $N_k \equiv N_k(t)$.

One challenge in this setting is defining a estimator representing the "best estimator in hindsight" that may be used to formulate a regret equation. Note that here Theorem 6.1 no longer applies and is not easily extended, in no small part due to the fact that $\hat{\mu}^*$ is not even unbiased since the individual estimates as a result of the division by N_k in each individual estimate: $\hat{\mu}_k = \frac{1}{N_k} \sum_{t=1}^{N_k} X_{k,t}$. Interestingly, in the uniform-cost setting the

simple *uniformly weighted* estimator

$$\hat{\mu}_t = \sum_{k=1}^{K} \lambda_k \hat{\mu}_k \quad \text{where} \quad \lambda_k = \frac{N_k}{\sum_{j=1}^{K} N_j}, \tag{6.6}$$

retains its unbiasedness, as the N_k values in the denominator are cancelled. However, this estimator is not particularly efficient or generally applicable (when we cannot assume $\sum_{k=1}^{K} N_k = n$ for deterministic n).

Ultimately, in the regret formulation for this section we will continue to use the estimator $\hat{\mu}^*$ from the previous section despite the fact that we are not able to show that it is best possible estimator in a formal sense. This choice is motivated by the fact that for large sample sizes the bias introduced by the division of the N_k values will be greatly diminished; Indeed, we later make this detail explicit by assuming the distribution each N_k concentrates about its mean. Under this assumption it the performance of this weighted estimator will be similar to that obtained in the deterministic setting. In particular we define

$$\hat{\mu}^* = \sum_{k=1}^{K} \lambda_k^* \hat{\mu}_k \quad \text{where} \quad \lambda_k^* := \frac{N_k/\sigma_k^2}{\sum_{j=1}^{K} N_j/\sigma_j^2},$$

which leads to the familiar regret formulation

$$R(\hat{\mu}, t) = t^2(L(\hat{\mu}) - L(\hat{\mu}^*)), \tag{6.7}$$

where $L(\hat{\mu}) = \mathbb{E}\left[(\hat{\mu} - \mu)^2\right]$. This formulation is the same as the regret in Eq. (6.3) with the exception that N_k are stopping times.

The proof technique in this section is generally the same as in Section 6.2 though we make use of the same assumptions, and lemmas, used in Chapter 5. Most critically we will we will assume that the random quantities of interest, $\hat{\sigma}_k^2$ and N_k, will concentrate around their means with high probability. In particular, we define the events

$$\xi_k := \left\{ |\hat{\sigma}_k^2 - \sigma_k^2| \leq \Delta_{\sigma_k^2}(N_k) \right\},$$
$$\mathcal{N}_k := \left\{ |\mathbb{E}[N_k] - N_k| \leq \Delta_{N_k} \right\},$$

with $\xi := \bigcap_{k=1}^{K} \xi_k$ and $\mathcal{N} := \bigcap_{k=1}^{K} \mathcal{N}_k$. Here ξ is unchanged from the previous section aside from changing the notation Δ_k to $\Delta_{\sigma_k^2}$; note also that $\Delta_k(n)$ is a monotonically decreasing function of n. Again, as we did in Chapter 5, we will require that $\Delta_{N_k} = c\sqrt{\mathbb{E}[N_k]}$ for some constant $c > 0$. In this context we define the upper confidence weighted estimator $\hat{\mu}_{\text{UCB-W}}$ in the same way as before (Eq. (6.5)) where σ_k^2 is replaced by $\hat{\sigma}_k^2 + \Delta_k(N_k)$.

Theorem 6.3. *For K sequences of i.i.d. random variables $X = \{X_{k,1}, ..., X_{k,N_k}\}_{1 \leq k \leq K}$ satisfying the same conditions expressed in Theorem 6.2 and where the number of samples, $\{N_k\}_{1 \leq k \leq K} \geq 2$, are chosen sequentially such that N_k is a stopping time w.r.t. X (satisfying the conditions of Lemma 4.2). Additionally, assuming that $\mathbb{P}(\mathcal{N}^C) = \eta$ and $\mathbb{P}(\xi^C) = \delta$ are bounded as $O(t^{-3/2})$, also that $\mathbb{E}[N_k] \leq C_f t$ for a fixed constant C_f, the UCB-W estimator satisfies*

$$R(\hat{\mu}_{\text{UCB-W}}, t) = \widetilde{O}(t^{1/2}) \,.$$

(The proof is similar to the deterministic case but requires bound for $|N_k - \mathbb{E}[N_k]|$; full proof is given in Appendix D.3.)

In summary, the additional complexities introduced by non-deterministic sample sizes are not enough to affect the regret bounds by more than constant factors. The most significant implication of this theory is the requirement that the number of samples allocated to each arm must be reasonably well concentrated around their means. Again, as we detailed in the previous chapter, this property can be achieved through careful parameterization of various bandit approaches but in general is nontrivial. That said, the empirical results detailed in the next section do suggest that this may be an artifact of the proof technique, especially for settings with large sample sizes.

6.4 Experimental Evaluation

We now explore the practical significance of the UCB-W estimator through empirical study. In particular, we are interested in examining the behaviour of this estimator in contrast to the simpler, and widely used, Graybill-Deal estimator. Additionally, we wish to examine any advantages that this estimator might confer on the bandit-based allocation procedures detailed in the previous chapters. Fortunately, for this latter question the empirical evaluation is straightforward: we need only recompute the final estimates for the various bandit algorithms on the same problems (using the same allocations). We may then compare and contrast the various effects of using weighted estimators in this context.

6.4.1 2-Arm Fixed Allocation Problem

We first consider the performance of the different weighted estimators on a synthetic experiment where samples are drawn from two scaled Bernoulli (SB) distributions according to a fixed allocation distribution. Specifically, for each $k \in \{1, 2\}$ we let $X_k \sim \text{SB}(\mu, \sigma_k^2)$ where $\mu = 0.5$ and the number of samples is given by $N_1 \sim 2 + \text{Binomial}(n-4, \gamma)$ $(n-4$

trials and fixed parameter γ), and $N_2 = 2 + n - N_1$ for $n = 1000$. Additionally, we fix the variances parameters $\sigma_1^2 = 0.25$ (Standard $p = 0.5$ Bernoulli) and $\sigma_2^2 = 0.025$ and consider different settings of $\gamma \in \{0.99, 0.9, 0.5, 0.1, 0.01\}$. Using these parameters we generate random variables N_k and $\{X_{k,1:N_k}\}_{k\in\{1,2\}}$ 25000 times and each time record the performance of the different weighted estimators as a function of n.

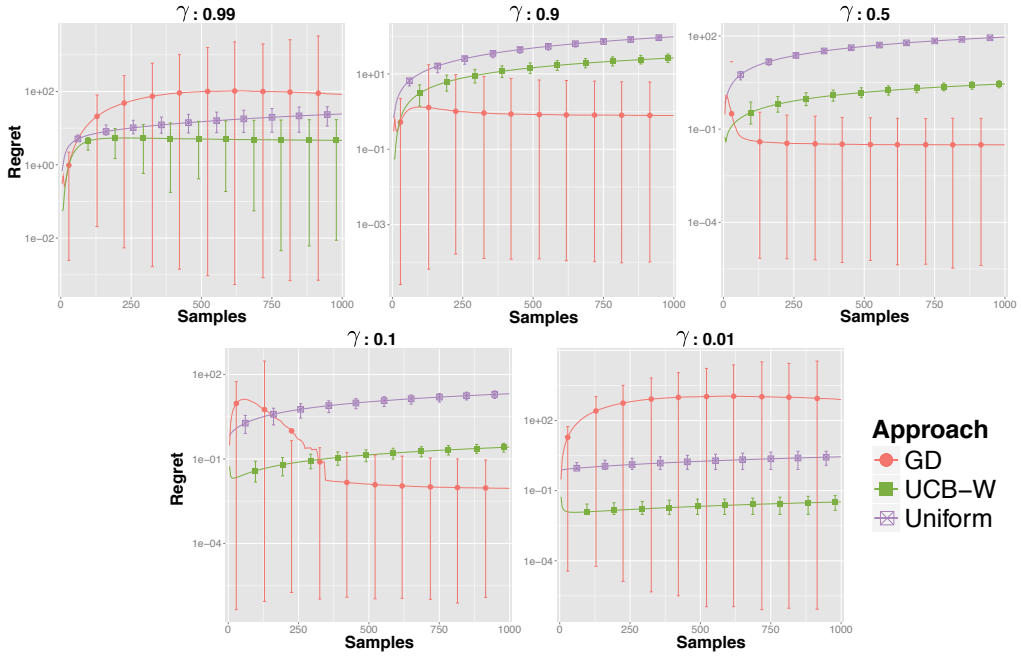

Figure 6.1: Average regret (y-axis, log-scale), and 99% empirical density regions, are plotted for each weighted estimator. Values are computed over 25000 simulations.

Figure 6.1 gives the regret for the the *uniformly weighted* estimator, Eq. (6.6), the *Graybill-Deal* (GD) estimator, Eq. (6.4), and the proposed *upper-confidence weighted* estimator (UCB-W), Eq. (6.5) for each setting of γ. Our first observation is that the UCB-W estimator consistently outperforms the uniformly weighted estimator by a considerable margin. Also, though it's not shown here, this margin grows rapidly as the difference in variances between the two distributions is increased.

In regards to the differences between the GD estimator and the UCB-W estimator, it is apparent that the GD estimator can significantly outperform the UCB-W estimator when each distribution is sampled sufficiently often. However, as suspected, this advantage comes at the cost of potentially disastrous performance when there are too few samples for either distribution. In particular, it appears that for the GD estimator the distribution for the regret values is bimodal: there is a relatively high probability the estimator will give low error and a low probability that the error is extremely high. In some cases the high errors incurred

in rare cases are enough to overshadow the lower errors in the more common cases on average. This is illustrated, in part, by the relationship between the 99% empirical density intervals and the averages in Fig. 6.1; specifically that at one point the average lies *outside* the 99% interval in the plot for $\gamma = 0.1$. The UCB-W estimator, on the other hand, gives up some performance benefits in the average case in order to prevent poor performance in the worst case, as intentioned. That said, in anecdotal comparisons we found that the UCB-W confidence bound suggested by the theory is quite conservative and much better empirical results may be obtained by scaling down these bounds. Of course the risks associated with scaling the confidence bounds can be hard to demonstrate empirically.

6.4.2 Bandit Experiments

The most promising aspect of the weighted estimators presented in this chapter is the positive implications for the Monte Carlo bandit approaches detailed in the preceding chapters; particularly for the non-uniform cost settings. Fortunately, it is straightforward to analyze the combined performance of the bandit approaches and a weighted estimator, like UCB-W, since the regret definition need not change significantly. In the earlier formulations the *best arm in hindsight* notion ensures that a single arm choice is used as a comparison while in this chapter we used the optimal weighted estimator. However, in the case where only one arm is chosen the optimally weighed estimator is identical to uniformly weighted estimator. As a result, we can simply substitute the MSE of the weighted estimator applied to the allocations taken by the bandit method $\mathcal{A}$ to arrive at the regret, or *normalized excess risk*, given by

$$R(\mathcal{A}, t) = t^2 \mathbb{E}\left[\hat{\mu}(\{X_{1:N_k(t)}\}_{1 \leq k \leq K}) - \frac{\delta_{k^*}\sigma_{k^*}^2}{t}\right],$$

where $\hat{\mu}$ is the weighted estimator of interest. Again, the regret for the uniform-cost setting can be expressed by letting $\delta_k = 1$ and $t = n$.

Uniform Cost Setting: Option Pricing

We begin by revisiting the option pricing experimental setup described in Section 4.5.2. Recall that in this experiment the bandit approaches were tasked with finding the best importance sampling proposal for approximating the expected value of a European caplet option. Additionally, this setup permitted the direct comparison with the popular population Monte Carlo (PMC) approach which selected proposal density through a resampling heuristic.

The experiment was conducted for one of the three parameterizations (strike price $s = 0.09$) and the resulting regret curves are given in Fig. 6.2. Recall, in Section 4.5.2

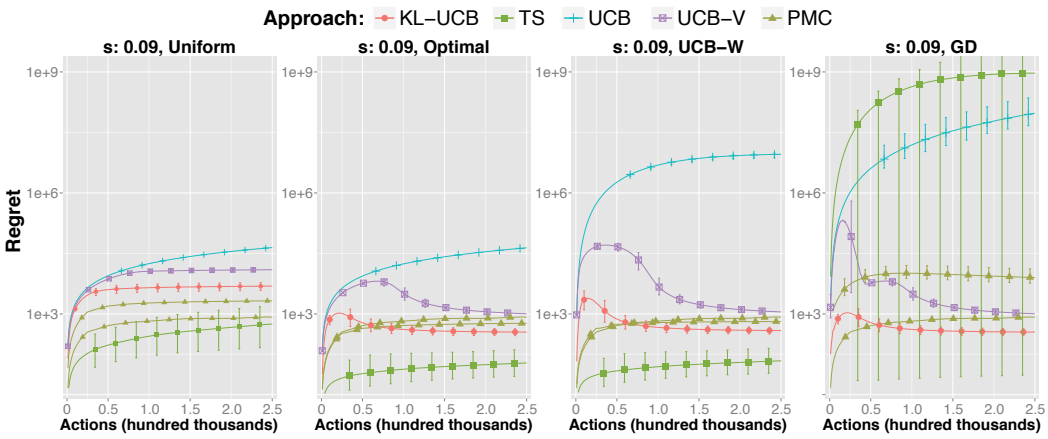

Figure 6.2: Results for the Cox-Ingersoll-Ross model problem setting with a strike price (s) of 0.09. The leftmost plot gives the regret for the uniformly weighted estimator and is identical to the rightmost plot in Fig. 4.3 with the exception of the log-scale. The regret for the *optimal* inverse-variance weighted estimator is next, followed by the UCB-W estimator and the Graybill-Deal (GD) estimator. Results are averaged over 10000 simulations and error bars delineate the are 99% empirical percentiles. The 2 curves for the PMC method give the performance for the "Rao-Blackwellized" (dashed) estimator as well as the weighted estimator in question (solid), that is using PMC only to allocate samples as a bandit method.

we observed that only the Thompson sampling approach was able to consistently outperform the PMC method; this is evident in the leftmost plot in the above figure. However, looking at the performance of the same bandit allocations using the *optimal* estimator (second plot) we see that regret reduces dramatically. In particular, the Thompson sampling approach goes from a 50% improvement over PMC to almost 1000%, the KL-UCB approach now significantly outperforms PMC, and the "Rao-Blackwellized" PMC estimator performs worse than the PMC sample allocations combined with the optimal weighted estimator.

The next question of whether these advantages are retained by the more practical UCB-W estimator is answered by the third plot. Here, it is clear that after a relatively short amount of sampling the performance of the UCB-W estimator closely matches that of the optimal estimator, with the exception of the UCB allocations. The poor performance on the UCB allocations is due to the fact that the samples are allocated almost uniformly across the arms which gives less time for the confidence regions in the UCB-W estimator to shrink.

The regret for the Graybill-Deal estimator, shown in the rightmost plot, is perhaps the most surprising. As we observed in the previous experiment, the GD estimator appears to do poorly when the arm-selection probabilities are imbalanced. Interestingly, the Thompson sampling approach has a tendency to eliminate arms after a small number of samples,

while this often leads to a better regret, this property can lead to highly suboptimal performance when combined with the GD estimator. Compared to UCB-W the GD estimator does exhibit improved performance for the KL-UCB and UCB-V allocations. However, improvements here are not likely enough to justify the risks associated with this estimator.

Non-Uniform Cost Setting: Adaptive AMCS

We now consider the performance of the weighted estimators for the non-uniform costs bandit setting starting with the adaptive AMCS experiments described in Section 5.3.1. Recall that for this particular experiment the bandit methods performed reasonably well and did not exhibit the drawbacks related to having multiple near-optimal arms.

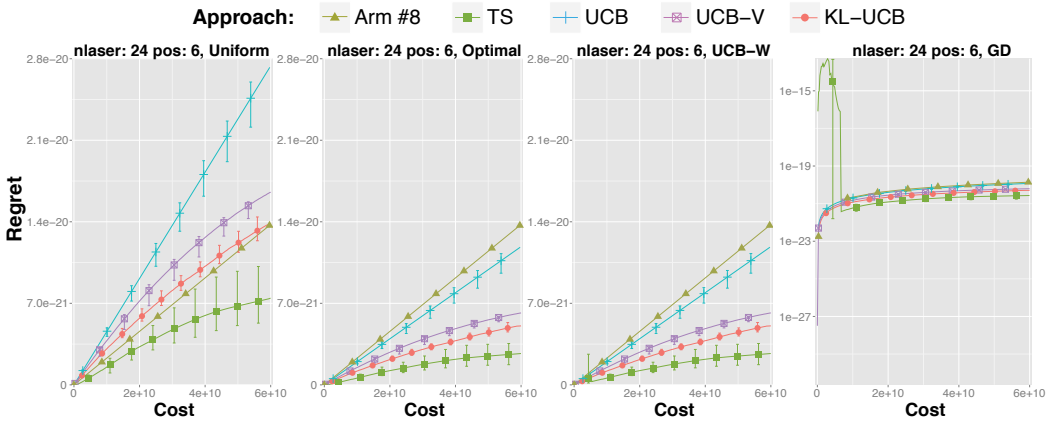

Figure 6.3: Regret curves for the weighted estimators for the adaptive AMCS setting detailed in Section 5.3.1, specifically the 24 laser, position #6, setting. The leftmost plot above shows the regret for the original (uniform) estimator and corresponds exactly to the rightmost plot in Fig. 5.2. From left to right the regret for the optimally weighted estimator and the UCB-W estimator are next followed finally by the Graybill-Deal estimator. This rightmost plot is in log-scale.

Nonetheless, the regret curve for the optimally weighted estimator shown in Fig. 6.3 indicate that the proper weighting results in a dramatic improvement over the uniform weighting. Using the regret for the single-arm approach (Arm #8) as a reference point, we can see that using the weighted estimators is as important, if not more important, than selecting the appropriate bandit approach for the task. Additionally, for this experiment we can see that the added confidence bound used in the UCB-W estimator does not introduce a noticeable amount of error since the regret is almost identical to the optimally weighted estimator. The Graybill-Deal estimator, however, appears to be prone to extremely poor performance for small time horizons. Specifically, in the rightmost plot, we observe that the regret is 5 orders of magnitude higher for Thompson sampling than the other methods early on. Although,

the regret does eventually recover to a value slightly lower than that of UCB-W for longer horizons.

Non-Uniform Cost Setting: Tuning Adaptive SMCS

We now consider the more exciting non-uniform bandit experiments featuring the SMCS algorithm. Recall that these experiments highlighted the potential for very poor performance when deploying bandit approaches in cases where the bandit arms are all close in terms of cost-adjusted-variance but very different in terms of variances (see Section 5.3.4). Specifically, we had observed that for a number of settings the performance of the bandit algorithms *was actually worse than pulling only the highest variance arm*. In the subsequent discussion the blame for this poor performance was placed on the inefficiencies of uniformly weighting the samples for each arm; we now test this hypothesis experimentally.

The regret curves in Fig. 6.4 show the performance of the different bandit approaches using the each of the weighted estimators for adaptive SMCS setting on the 4 dimensions mixture distribution described in Section 5.3.2. The leftmost plot in this figure shows the regret when using the uniformly weighted estimator, where none of the bandit methods were able to perform anywhere near as well as sampling from a single suboptimal arm (Arm #10).

The adjacent plot tells a much different story, here we can see that by optimally weighting the samples the same allocations have a regret that is two orders of magnitude lower than their previous values (except UCB) and considerably better than the constant arm. In the next plot we can see that these gains transfer almost exactly as well when using the UCB-W estimator. There is a small difference in the first few hours of sampling for some of the approaches, but this difference disappears relatively quickly.

Additionally, looking at the regret for the Graybill-Deal (GD) estimator in the final plot we can see, again, that the estimator does not combine well with the TS algorithm. This is another testament to the risk associated with weighting samplers from the sample variance computed from only a handful of samples.

In addition to the adaptive SMCS setting similar disappointing performance was observed when tuning the annealed importance sampling (AIS) method in Section 5.3.3. Recall that the setup in for these simulations was the same as above with the exception that the bandit approaches are attempting to find the optimal annealing schedule as opposed to population size. The results for the weighted estimator simulations are summarized in Fig. 6.5. These results once again demonstrate the considerable advantages of using weighted es-

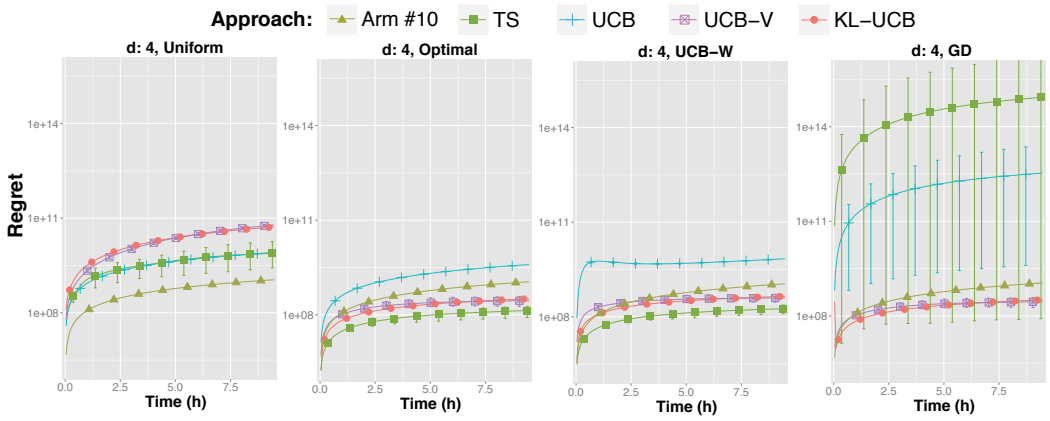

Figure 6.4: Regret curves for the bandit approaches using the different weighted estimators for the adaptive SMCS setting for the 4-dimensional Gaussian mixture as described in Section 5.3.2. The leftmost plot gives the regret for the uniform estimator and corresponds to the middle plot in Fig. 5.4, in log scale. The regret when using the optimally weighted estimator, the UCB-W estimator, and the Graybill-Deal estimator are also shown.

timators in this setting. Here, by applying the optimally weighted estimator the bandit approaches go from being up to two orders of magnitude worse than the constant arm to consistently outperforming it by a factor of 2x. Of course, UCB is again the one exception, which is due to the fact that it is not aggressively selecting the optimal arms.

Again the performance of the UCB-W estimator is closely matched to the that of optimal estimator though some amount of regret is incurred by the conservative estimates on sample variances. As with previous examples the rightmost plot for the GD estimator makes it clear that the risks associated with this estimator makes is essentially untenable in practice.

6.5 Discussion

In this chapter we considered the problem of combining multiple unbiased estimators into a common estimate. We have shown that taking the convex combination of these estimators weighted inversely proportional to their respective variances results in both the minimum variance unbiased estimator as well as a minimax estimator. Since this estimator is not applicable in practice we formulated an alternative approach, UCB-W, which uses an upper-confidence bound on the sample variance. We prove that this approach has a regret, in respect to the optimal weighting, of $\widetilde{O}(\sqrt{n})$ when the samples are chosen deterministically and go on to show a similar result in the case where the samples are stochastic. This latter result can then be applied to the non-uniform cost bandit setting discuseed in the preceding

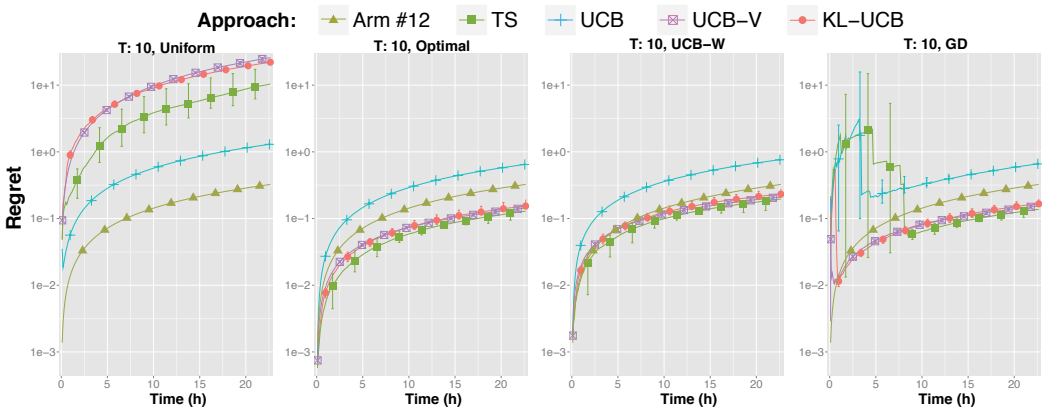

Figure 6.5: Regret curves for the bandit approaches using the different weighted estimators for tuning AIS for a logistic regression model with T=10 training examples, as described in Section 5.3.3. The leftmost plot gives the regret for the uniform estimator and corresponds to the middle plot in Fig. 5.7, in log scale. The regret when using the optimally weighted estimator, the UCB-W estimator, and the Graybill-Deal estimator are also shown.

chapter to recover the $\widetilde{O}(\sqrt{t})$ rate when the optimal arm cannot be assumed to be unique. Additionally, the empirical results for the UCB-W estimator provide compelling evidence for the use of this technique in both the uniform and non-uniform cost bandit settings. In particular, the results indicate the use of this estimator is possibly even more critical than selecting the ideal underlying bandit approach.

With respect the future development of the methods presented in this chapter we feel that the most straightforward next steps would be to expand the theory to encompass the more powerful confidence bound formulations, such as the empirical Bernstein bound used in UCB-V (see Appendix D.4.2) as well as the KL bound of KL-UCB. Unfortunately, the proof techniques used in this chapter make extensive use of the fact that the UCB-style bound is a deterministic function of the number of samples N_k and, therefore, may be difficult to adapt to these bounds. However, given the observed gap in the performance between the UCB-W estimator and the optimally weighted estimator, as well as the gap in performance between UCB and UCB-V, we feel there is a lot to be gained through research in this direction.

Chapter 7

Concluding Remarks

In this thesis we have introduced a number of novel ways in which Monte Carlo integration approaches can adapt their behaviour automatically to target integrand. Our first approach, antithetic Markov chain sampling, achieves this adaptation though the use of predefined Markov chains that terminate according to given stopping rules. Ultimately, this formulation allows one to exploit detailed aspects the integrand. For example, minimizing the computation spent evaluating the integrand in insignificant regions (values below threshold), and by averaging-out high frequency *peaks*, we find that the approximation error can be greatly reduced. Additionally, because this approach does not introduce correlations between sequential samples it is straightforward to apply to more complex scenarios, for instance sampling within parallel architectures. The utility of this approach was demonstrated on complex machine learning tasks that challenge existing methods.

The bandit-based Monte Carlo sampling approaches presented in the subsequent chapters take a different, but complimentary, approach to adaptation. By formulating the adaptive Monte Carlo problem as a sequential allocation task we were able to prove a reduction to the classic stochastic bandit problem. As a result, we are able to immediately take advantage of the rich body of literature and practical algorithms developed for this setting. The resulting bandit-based adaptive Monte Carlo methods are not only practically significant but bring with them strong finite-time performance guarantees, a rarity in the adaptive Monte Carlo literature.

The non-uniform costs extension presented in the following chapter was natural next step for this bandit setup as it permitted the construction of more sophisticated adaptive algorithms. In particular, allocation algorithms that can tune the parameters for AMCS or SMCS approaches. It was shown that it is straightforward to apply existing bandit methods to this extended formulation through a simple double-sampling trick. An upper bound

on the regret of these bandit approaches, with a rate of $\widetilde{O}(\sqrt{t})$, was proven under mild technical conditions. Empirical results uncovered generally positive results for tuning an AMCS sampler but somewhat negative results for tuning SMCS approaches. These negative results stemmed from rather unique aspects of SMCS which uncovered limitations in how the sampled values were combined into a final estimate.

This somewhat unfortunate result motivated the efforts in the subsequent chapter surrounding the general problem of combining unbiased estimators into a single estimate. Here, it was shown that upper-confidence bound techniques, similar to those used in the multi-armed bandit literature, can be used to construct practical and theoretically sound estimators. The resulting UCB-W estimator was shown to have a regret of $\widetilde{O}(\sqrt{t})$ in settings where the samples were allocated stochastically or deterministically. This result covered an important edge-case in the non-uniform cost version of the bandit-based adaptive Monte Carlo framework uncovered previously. In addition to the theoretical improvements this estimator was shown to offer significant empirical improvements in both the uniform and non-uniform cost settings.

In summary, this thesis has presented a number of unique ways to automatically adapt Monte Carlo integration approaches to the targeted integrand. We feel that these approaches are both practically useful and representative of an interesting area for future research.

Bibliography

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*.

Agrawal, S. and Goyal, N. (2012). Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory (COLT)*.

Ahn, S., Korattikara, A., and Welling, M. (2012). Bayesian posterior sampling via stochastic gradient fisher scoring. In *International Conference on Machine Learning (ICML)*.

Andrieu, C., Moulines, É., et al. (2006). On the ergodicity properties of some adaptive mcmc algorithms. *The Annals of Applied Probability*, 16(3):1462–1505.

Antos, A., Grover, V., and Szepesvári, C. (2010). Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411(29):2712–2728.

Arouna, B. (2004). Adaptive Monte Carlo technique, a variance reduction technique. *Monte Carlo Methods and Applications*.

Audibert, J., Munos, R., and Szepesvári, C. (2009). Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902.

Audibert, J.-Y. and Bubeck, S. (2010). Regret bounds and minimax policies under partial monitoring. *The Journal of Machine Learning Research*, 11:2785–2836.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (2002b). The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32:48–77.

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Badanidiyuru, A., Kleinberg, R., and Slivkins, A. (2013). Bandits with knapsacks. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 207–216. IEEE.

Bardenet, R., Doucet, A., and Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *International Conference on Machine Learning (ICML)*.

Beskos, A., Jasra, A., and Thiery, A. (2013). On the convergence of adaptive sequential monte carlo methods. *arXiv preprint arXiv:1306.6462*.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. springer New York.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. (2013). Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2011). X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695.

Burnetas, A. and Katehakis, M. (1996). Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17:122–142.

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459.

Cappé, O., Garivier, A., Maillard, O., Munos, R., and Stoltz, G. (2013). Kullback-Leibler upper confidence bounds for optimal sequential decision making. *Annals of Statistics*, 41(3):1516–1541.

Cappe, O., Guillin, A., Marin, J.-M., Robert, C. P., and Robertyz, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13:907–929.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation*, 146(1):2–7.

Carpentier, A. (2012). *De l'échantillonnage optimal en grande et petite dimension (Optimal sampling in large and small dimesions)*. PhD thesis, Universite Lille I Nord de France.

Carpentier, A. and Munos, R. (2011). Finite-time analysis of stratified sampling for Monte Carlo. In *Advances in Neural Information Processing Systems (NIPS)*.

Carpentier, A., Munos, R., and Antos, A. (2014). Adaptive strategy for stratified monte carlo sampling. *Journal of Machine Learning Research*.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Chapelle, O. and Li, L. (2011). An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems (NIPS)*.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Chow, Y. S. and Robbins, H. (1963). On optimal stopping rules. *Probability Theory and Related Fields*, 2(1):33–49.

Cox, D. R., Cox, D. R., Cox, D. R., and Cox, D. R. (1962). *Renewal theory*, volume 4. Methuen London.

Cox, J., Ingersoll Jr, J., and Ross, S. (1985). A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society*, pages 385–407.

Del Moral, P. and Doucet, A. (2002). Sequential Monte Carlo samplers. *arXiv preprint cond-mat/0212648*.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:411–436.

Doob, J. (1953). *Stochastic Processes*. Wiley.

Douc, R., Guillin, A., Marin, J., and Robert, C. (2007a). Convergence of adaptive mixtures of importance sampling schemes. *Annals of Statistics*, 35:420–448.

Douc, R., Guillin, A., Marin, J., and Robert, C. (2007b). Minimum variance importance sampling via population Monte Carlo. *ESAIM: Probability and Statistics*, 11:427–447.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Earl, D. J. and Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916.

Etoré, P. and Jourdain, B. (2010). Adaptive optimal allocation in stratified sampling methods. *Methodology and Computing in Applied Probability*, 12(3):335–360.

Garivier, A. (2013). Informational confidence bounds for self-normalized averages and applications. IEEE Information Theory Workshop p.489-493.

Gelman, A. and Meng, X.-L. (1997). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling.

Ghosh, J. K., Delampady, M., and Samanta, T. (2006). *An introduction to Bayesian analysis*. Springer New York.

Girolami, M. and Calderhead, B. (2011). Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.

Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Springer.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*.

Graybill, F. A. and Deal, R. (1959). Combining unbiased estimators. *Biometrics*, 15(4):543–550.

Grover, V. (2009). Active learning and its application to heteroscedastic problems. Master's thesis, University of Alberta, Department of Computing Science.

Gut, A. (2005). *Probability: a graduate course?* Springer.

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2011). Inference for lévy-driven stochastic volatility models via adaptive sequential monte carlo. *Scandinavian Journal of Statistics*, 38(1):1–22.

Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling: An asymptotically optimal finite time analysis. In *Algorithmic Learning Theory (ALT)*.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics*, 5(1):1–25.

Kuss, M. and Rasmussen, C. (2005). Assessing approximate inference for binary gaussian process classification. *The Journal of Machine Learning Research*, 6:1679–1704.

Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22.

Lehmann, E. L. and Casella, G. (1998). *Theory of point estimation*, volume 31. Springer.

Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer.

Liu, J. S. and Chen, R. (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

Lizzote, D. (2008). *Practicle Bayesian Optimization*. PhD thesis, University of Alberta.

Ma, T., Ye, R., and Jia, L. (2011). Finite-sample properties of the graybill–deal estimator. *Journal of Statistical Planning and Inference*, 141(11):3675–3680.

Maurer, A. and Pontil, M. (2009). Empirical bernstein bounds and sample variance penalization. *Conference on Learning Theory (COLT)*.

May, B. and Leslie, D. (2011). Simulation studies in optimistic Bayesian sampling in contextual-bandit problems. Technical report, 11: 02, Statistics Group, Department of Mathematics, University of Bristol.

Mitra, P. K. and Sinha, B. K. (2007). On some aspects of estimation of a common mean of two independent normal populations. *Journal of statistical planning and inference*, 137(1):184–193.

Neal, R. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical report, University of Toronto.

Neal, R. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and computing*, 6(4):353–366.

Neal, R. (2001). Annealed importance sampling. Technical report, University of Toronto.

Neal, R. (2003). Slice sampling. *Annals of statistics*, pages 705–741.

Neal, R. (2005). Estimating ratios of normalizing constants using linked importance sampling. Technical report, University of Toronto.

Neal, R. (2011). *Handbook of Markov Chain Monte Carlo*, chapter MCMC using Hamiltonian dynamics, pages 113–162. Chapman & Hall / CRC Press.

Neufeld, J., Bowling, M., and Schuurmans, D. (2015). Variance reduction via antithetic Markov chains. In *Artificial Intelligence and Statistics (AISTATS)*.

Neufeld, J., György, A., Schuurmans, D., and Szepesvári, C. (2014). Adaptive Monte Carlo via bandit allocation. In *International Conference on Machine Learning (ICML)*.

Newton, M. A. and Raftery, A. E. (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society Series B Methodological*, 56(1):3–48.

Oh, M. and Berger, J. (1993). Integration of multimodal functions by monte carlo importance sampling. *Journal of the American Statistical Association*, 88:450–456.

Powell, M. J. D. and Swann, J. (1966). Weighted uniform sampling, a Monte Carlo technique for reducing variance. *IMA Journal of Applied Mathematics*.

Richard, J.-F. and Zhang, W. (2007). Efficient high-dimensional importance sampling. *Journal of Econometrics*, 141(2):1385–1411.

Robbins, H. (1952). Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535.

Robert, C. P. (2010). *Handbook of Computational Statistics (revised)*, chapter 11, Bayesian Computational Methods, http://arxiv.org/abs/1002.2702. Springer.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods*. Springer-Verlag New York.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, 18(2):349–367.

Schäfer, C. and Chopin, N. (2013). Sequential monte carlo on large binary sampling spaces. *Statistics and Computing*, 23(2):163–184.

Southey, F., Schuurmans, D., and Ghodsi, A. (2002). Regularized greedy importance sampling. In *Advances in Neural Information Processing Systems (NIPS)*.

Thompson, W. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. MIT Press.

Appendix

Appendix A

AMCS

A.1 Proof of Lemma 3.1

The proof works backwards from μ, uses the symmetric properties of k to swap arguments, then swaps the order of integration, in particular we have

$$\begin{aligned}
\mu &= \int h(x')\pi(x')dx' \\
&= \int h(x')\pi(x') \int k(x', x)dxdx' \\
&= \int h(x')\pi(x') \int k(x, x')dxdx' \qquad \text{(symmetry of } k\text{)} \\
&= \int \int h(x')\pi(x')k(x, x')dx'dx \qquad \text{(Fubini's theorem)} \\
&= \int \int \frac{h(x')\pi(x')}{\pi_0(x)} k(x, x')\pi_0(x)dx'dx \\
&= \mathbb{E}\left[\frac{h(X')\pi(X')}{\pi_0(X)}\right].
\end{aligned}$$

Above we observe that π and k are valid densities (Definition 2.1) which ensures that, by Fubini's theorem, the iterated integrals equal the joint integral and the order may be swapped.

A.2 Proof of Lemma 3.2

To prove (i) we show that the integral reduces to a infinite telescoping sum and make use of Assumption 3.1 to cancel the final term. In particular, we let

$$Z := \int \sum_{n,m\geq 1} \gamma(x^{(-n)}, ..., x^{(m)}, n, m|x^{(0)})dx_{-0}^{(-n:m)},$$

then, considering only the positive chain, for any fixed $m \geq 1$ we have

$$\begin{aligned}
&\int (1 - \alpha^+(x^{(m-1)}, x^{(m)}))k^+(x^{(m-1)}, x^{(m)}) \\
&\qquad \times \prod_{j=1}^{m-1} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)})dx^{(1:m)} \\
&= \int \prod_{j=1}^{m-1} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)})dx^{(1:m-1)} \\
&\qquad - \int \prod_{j=1}^{m} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)})dx^{(1:m)} \\
&=: s_{m-1} - s_m,
\end{aligned}$$

where $s_0 := \int k^+(x^{(0)}, x^{(1)})dx^{(1)}$. Taking the infinite sum then recovers

$$\lim_{m' \to \infty} \sum_{m=1}^{m'} s_{m-1} - s_m = s_0,$$

since, by Assumption 3.1, $\lim_{m' \to \infty} s_{m'} = 0$. We may then observe that $s_0 = 1$ since k is a valid conditional density. Replicating the same steps for the negative chain, using the notation s'_n, we have

$$Z = (\sum_{n \geq 1} s'_{n-1} - s'_n)(\sum_{m \geq 1} s_{m-1} - s_m) = 1$$

as desired.

To prove (ii) we first observe that since $x^{(-n)} = x'^{(-n')}, x^{(1-n)} = x'^{(1-n')}, ..., x^{(m)} = x'^{(m')}$, and $m+n = m'+n'$ where $m, n, m', n' \geq 1$, then $x'^{(0)} = x^{(k)}$ for some $n < k < m$. The proof is therefore a matter of shifting $|k|$ terms from one product to the other in the definition of γ in Eq. (3.3). In particular we have

$$
\begin{aligned}
\gamma(x^{(-n)}, ..., x^{(m)}, n, m | x^{(0)}) &= (1 - \alpha^+(x^{(m-1)}, x^{(m)}))k^+(x^{(m-1)}, x^{(m)}) \\
&\times \prod_{j=1}^{m-1} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)}) \\
&\times (1 - \alpha^-(x^{(1-n)}, x^{(-n)}))k^-(x^{(1-n)}, x^{(n)}) \\
&\times \prod_{j=-1}^{1-n} \alpha^-(x^{(j+1)}, x^{(j)})k^-(x^{(j+1)}, x^{(j)}) \\
&= (1 - \alpha^+(x^{(m-1)}, x^{(m)}))k^+(x^{(m-1)}, x^{(m)}) \\
&\times \prod_{j=1-k}^{m-1} \alpha^+(x^{(j-1)}, x^{(j)})k^+(x^{(j-1)}, x^{(j)}) \\
&\times (1 - \alpha^-(x^{(1-n)}, x^{(-n)}))k^-(x^{(1-n)}, x^{(n)}) \\
&\times \prod_{j=-1-k}^{1-n} \alpha^-(x^{(j+1)}, x^{(j)})k^-(x^{(j+1)}, x^{(j)}) \\
&= \gamma(x'^{(-n')}, ..., x'^{(m')}, n', m' | x'^{(0)}),
\end{aligned}
$$

where the second equality follows from the fact that (k^+, α^+) and (k^-, α^-) are jointly symmetric (Definition 3.1), and the last equality from renaming variables according to: $x^{(-n)} = x'^{(-n')}, x^{(1-n)} = x'^{(1-n')}, ..., x^{(m)} = x'^{(m')}$.

A.3 Proof of Lemma 3.3

Recall that we have variables $(X_0, ..., X_L, L)$, where $X_i \in \mathbb{R}^d$, $L \in \mathbb{N}$, and $L \geq 2$, distributed according to some joint density γ and a random variable $J \sim \text{Uniform}(\{1, ..., L-1\})$ the variable $X_J = \sum_{j=0}^{l} \mathbb{I}\{J = j\} X_j$. We then have

$$
\begin{aligned}
\mathbb{P}(X_J \in A) &= \sum_{l \geq 2} \mathbb{P}(X_J \in A, L = l) \\
&= \sum_{l \geq 2} \sum_{j=1}^{l-1} \mathbb{P}(X_j \in A, J = j | L = l) \, \mathbb{P}(L = l) \\
&= \sum_{l \geq 2} \sum_{j=1}^{l-1} \mathbb{P}(X_j \in A | L = l) \, \mathbb{P}(J = j | L = l) \, \mathbb{P}(L = l) \, ,
\end{aligned}
$$

where we have used the fact that X_j and J are independent given L. This equation then becomes

$$\begin{aligned}
\mathbb{P}(X_J \in A) &= \sum_{l\geq 2} \frac{1}{l-1} \sum_{j=1}^{l-1} \mathbb{P}(X_j \in A, L = l) \\
&= \sum_{l\geq 2} \frac{1}{l-1} \sum_{j=1}^{l-1} \int_A \int \gamma(x_1, ..., x_l, l) dx_{1:l}^{-j} dx_j \\
&= \sum_{l\geq 2} \frac{1}{l-1} \sum_{j=1}^{l-1} \int_A \gamma_j(x, l) dx \\
&= \int_A \left(\sum_{l\geq 2} \frac{1}{l-1} \sum_{j=1}^{l-1} \gamma_j(x, l) \right) dx,
\end{aligned}$$

which gives the desired p.d.f.

A.4 Proof of Lemma 3.4

The key observation used to prove Lemma 3.4 is that for any trajectory $\bar{x} := (x^{(-n)}, ..., x^{(m)})$ with $x^{(0)} = y$ and $x^{(j)} = y'$, for any $-n \leq j \leq m, j \neq 0$, there exists exactly one "shifted" trajectory $\bar{x}' := (x'^{(-n')}, ..., x'^{(m')})$ where $x^{(-n)} = x'^{(-n')}$, $x^{(1-n)} = x'^{(1-n')}$, ..., $x^{(-m)} = x'^{(-m')}$ as well as $n' = n + j$, $m' = m - j$, $x'^{(0)} = y'$, and $x'^{(j)} = y$. Further, thanks to Lemma 3.2 we know that $\gamma(\bar{x} \setminus x^{(0)}, n, m | x^{(0)} = y) = \gamma(\bar{x}' \setminus x'^{(0)}, n', m' | x'^{(0)} = y')$. The proof therefore amounts to taking the conditional density in Eq. (3.4), remapping the indicies, and swapping out each trajectory with its "shifted" version. In particular, recalling the notation $x(n, m, x, j) = (x^{(-n)}, ..., x^{(m)})$, for $n, m \geq 1$, $-n \leq j \leq m$, and $x^{(j)} = x$, we have that

$$\begin{aligned}
\gamma(x|x') &= \sum_{l\geq 2} \frac{1}{l-1} \sum_{m=1}^{l-1} \sum_{j=m-l+1}^{m-1} \int \gamma(x(l-m, m, x, j), l-m, m | x^{(0)} = x') dx_{-\{j,0\}}^{(m-l:m)} \\
&= \sum_{l\geq 2} \frac{1}{l-1} \sum_{m=1}^{l-1} \sum_{j=m-l+1}^{m-1} \int ... \qquad \text{(A.1)} \\
&\qquad \gamma(x(l-m+j, m-j, x', -j), l-m+j, m-j | x^{(0)} = x) dx_{-\{j,0\}}^{(m-l:m)}. \qquad \text{(A.2)}
\end{aligned}$$

At this point we need only rearrange the inner two summations. To do so, we let $s(m, j) := \int \gamma(x(l-m+j, m-j, x', -j), l-m+j, m-j | x^{(0)} = x) dx_{-\{j,0\}}^{(m-l:m)}$. We may now

rearrange the summations as follows

$$\begin{aligned}
\sum_{m=1}^{l-1}\sum_{j=m-l+1}^{m-1} s(m,j) &= \sum_{m=1}^{l-1}\sum_{j=l+1}^{-1} s(m,j+m) \\
&= \sum_{j'=-1}^{1-l}\sum_{m'=1-l}^{1} s(-j',-m'-j') \quad \text{(using: } j=-m', m=-j') \\
&= \sum_{m'=1-l}^{1}\sum_{j'=-1}^{1-l} s(-j',-m'-j') \\
&= \sum_{m'=1}^{1-l}\sum_{j'=1-l}^{-1} s(-j',-m'-j') \\
&= \sum_{m'=1}^{1-l}\sum_{j'=m'-l+1}^{m'-1} s(m'-j',-j') \\
&= \sum_{m'=1}^{1-l}\sum_{j'=m'-l+1}^{m'-1} \int \gamma(x(l-m',m',x',j'),l-m',m'|x^{(0)}=x)dx_{-\{j,0\}}^{(m-l:m)}.
\end{aligned}$$

Substituting this expression into Eq. (A.2) concludes the proof.

Appendix B

Monte Carlo Bandits

B.1 Proof of Lemma 4.2

Lemma B.1 (Optional Sampling)**.** *Let $(X_t)_{t\in\mathbb{N}}$ be a sequence of i.i.d. random variables, and $(X'_t)_{t\in\mathbb{N}}$ be its subsequence such that the decision whether to include X_t in the subsequence is independent of future values in the sequence, i.e., X_s for $s \geq t$. Then the sequence $(X'_t)_{t\in\mathbb{N}}$ is an i.i.d. sequence with the same distribution as $(X_t)_{t\in\mathbb{N}}$.*

Proof. See Theorem 5.2 in Chapter III on page 145 of (Doob, 1953). □

proof of Lemma 4.2. Since we can always employ the substitution $X' = X - \mu$ we may, without loss of generality, consider the case when $\mu = 0$, we then observe that

$$\begin{aligned}
\mathbb{E}[S_{k,n}S_{j,n}] &= \mathbb{E}[(\mathbb{I}\{I_n = k\}Y_n + S_{k,n-1})(\mathbb{I}\{I_n = j\}Y_n + S_{j,n-1})] \\
&= \mathbb{E}\Big[\cancel{\mathbb{I}\{I_n = k\}\mathbb{I}\{I_n = j\}Y_n^2}\Big] + \mathbb{E}[\mathbb{I}\{I_n = k\}Y_n S_{j,n-1}] \\
&\quad + \mathbb{E}[\mathbb{I}\{I_n = j\}Y_n S_{k,n-1} + S_{k,n-1}S_{j,n-1}]\,.
\end{aligned}$$

By considering the conditional expectation w.r.t. the history up to Y_n (including I_n), i.e., w.r.t. $\mathcal{F}_{n-1} = \sigma(Y_1, \ldots, Y_{n-1}, I_1, \ldots, I_n)$, we have

$$\begin{aligned}
\mathbb{E}[S_{k,n}S_{j,n}] &= \mathbb{E}[\mathbb{E}[S_{k,n}S_{j,n}|\mathcal{F}_{n-1}]] \\
&= \mathbb{E}[(\mathbb{I}\{I_n = k\}S_{j,n-1} + \mathbb{I}\{I_n = j\}S_{k,n-1})\,\mathbb{E}[Y_n|\mathcal{F}_{n-1}]] + \mathbb{E}[S_{k,n-1}S_{j,n-1}]\,.
\end{aligned}$$

Now, since $I_n \in \{1, \ldots, K\}$ and I_n is $\mathcal{F}_{n-1}$-measurable,

$$\mathbb{E}[Y_n|\mathcal{F}_{n-1}] = \sum_{k=1}^{K} \mathbb{I}\{I_n = k\}\mathbb{E}\big[X_{k,T_k(n)}|\mathcal{F}_{n-1}, I_n = k\big]\,.$$

Further, by Lemma B.1, $(X_{k,T_k(n)})_n$ is an i.i.d. sequence, sharing the same distribution as $(X_{k,t})_t$. Since $\sigma(X_{k,T_k(n)})$ is independent of $\sigma(Y_1, \ldots, Y_{n-1}, I_1, \ldots, I_{n-1}, I_n, I_n = k)$,

$\mathbb{E}\big[X_{k,T_k(n)}|\mathcal{F}_{n-1}, I_n = k\big] = \mathbb{E}\big[X_{k,T_k(n)}\big] = \mu = 0$. Therefore,

$$\mathbb{E}[S_{k,n}S_{j,n}] = \mathbb{E}[S_{k,n-1}S_{j,n-1}] = \ldots = \mathbb{E}[S_{k,0}S_{j,0}] = 0.$$

Which concludes the proof. □

Appendix C

Monte Carlo Bandits With Costs

C.1 Proof for Lemma 5.1

Proof. Let $b = c\sqrt{\mathbb{E}[N]}$. First we prove the upper bound. We have

$$\begin{aligned}
\mathbb{E}\big[D^2\big] &\leq \mathbb{E}\big[D^2\mathbb{I}\{N \geq \mathbb{E}[N] - b, b \leq \tfrac{1}{2}\mathbb{E}[N]\}\big] + \mathbb{E}\big[D^2\mathbb{I}\{N < \mathbb{E}[N] - b\}\big] + \mathbb{E}\big[D^2\mathbb{I}\{b > \tfrac{1}{2}\mathbb{E}[N]\}\big] \\
&\leq \mathbb{E}\left[\frac{(S - N\mu)^2}{(\mathbb{E}[N] - b)^2}\mathbb{I}\{b \leq \tfrac{1}{2}\mathbb{E}[N]\}\right] + d^2\mathbb{P}(N < \mathbb{E}[N] - b) + d^2\mathbb{I}\{b > \tfrac{1}{2}\mathbb{E}[N]\} \\
&\leq \frac{\mathbb{E}\big[(S - N\mu)^2\big]}{(\mathbb{E}[N] - b)^2}\mathbb{I}\{b \leq \tfrac{1}{2}\mathbb{E}[N]\} + d^2\mathbb{P}(N < \mathbb{E}[N] - b) + d^2\mathbb{I}\{b > \tfrac{1}{2}\mathbb{E}[N]\}\,.
\end{aligned}$$

Noting that $1/(1-x) \leq 1 + 2x$ when $0 \leq x \leq 1/2$, we get that

$$\frac{1}{\mathbb{E}[N] - b}\mathbb{I}\{b \leq \tfrac{1}{2}\mathbb{E}[N]\} = \frac{1}{\mathbb{E}[N]}\,\frac{1}{1 - \frac{b}{\mathbb{E}[N]}}\mathbb{I}\{b \leq \tfrac{1}{2}\mathbb{E}[N]\} \leq \frac{1}{\mathbb{E}[N]}\left(1 + 2\frac{b}{\mathbb{E}[N]}\right) = \frac{1}{\mathbb{E}[N]}\left(1 + 2\frac{c}{\sqrt{\mathbb{E}[N]}}\right).$$

Putting things together, we get the desired upper bound.

The lower bound is proved in a similar fashion. To simplify notation, let $\mathcal{N}$ denote the event $\{N \leq \mathbb{E}[N] + b\}$. Then,

$$\begin{aligned}
\mathbb{E}\big[D^2\big] &\geq \mathbb{E}\left[\left(\frac{(S - N\mu)}{N}\right)^2\mathbb{I}\{\mathcal{N}\}\right] \\
&\geq \mathbb{E}\left[\frac{(S - N\mu)^2}{(\mathbb{E}[N] + b)^2}\mathbb{I}\{\mathcal{N}\}\right] \\
&= \mathbb{E}\left[\frac{(S - N\mu)^2}{(\mathbb{E}[N] + b)^2}\right] - \mathbb{E}\left[\frac{(S - N\mu)^2}{(\mathbb{E}[N] + b)^2}\mathbb{I}\{\mathcal{N}^C\}\right] \\
&\geq \frac{\mathbb{E}[N]^2}{(\mathbb{E}[N] + b)^2}\frac{\mathbb{E}\big[(S - N\mu)^2\big]}{\mathbb{E}[N]^2} - \mathbb{E}\left[\frac{(S - N\mu)^2}{(\mathbb{E}[N] + b)^2}\mathbb{I}\{\mathcal{N}^C\}\right] \\
&\geq \frac{\mathbb{E}[N]^2}{(\mathbb{E}[N] + b)^2}\frac{\mathbb{E}\big[(S - N\mu)^2\big]}{\mathbb{E}[N]^2} - \frac{\sqrt{\mathbb{E}[N^4]}d^2\mathbb{P}\big(\mathcal{N}^C\big)}{(\mathbb{E}[N] + b)^2} \\
&\geq \left(1 - \frac{2b}{\mathbb{E}[N]}\right)\frac{\mathbb{E}\big[(S - N\mu)^2\big]}{\mathbb{E}[N^2]} - \frac{\sqrt{\mathbb{E}[N^4]}d^2\mathbb{P}\big(\mathcal{N}^C\big)}{\mathbb{E}[N]^2}\,,
\end{aligned}$$

where the second last inequality follows from Cauchy-Schwarz and in the last inequality we used

$$\frac{\mathbb{E}[N]^2}{(\mathbb{E}[N]+b)^2} = \left(1 - \frac{b}{\mathbb{E}[N]+b}\right)^2 \geq 1 - \frac{2b}{\mathbb{E}[N]+b}$$

and $1/(\mathbb{E}[N]+b) \leq 1/\mathbb{E}[N]$. □

C.2 Proof for Lemma 5.2

Proof. We start with the proving the upper bound on $\mathbb{E}[N_k(t)]$ in (5.8). Define $\hat{D}_{k,m} = \min\{D_{k,m}, \beta\}$ and let $\hat{N}_k(t)$ be the smallest integer such that $\sum_{m=1}^{\hat{N}_k(t)} \hat{D}_{k,m} \geq t$. Then, clearly $\hat{N}_k(t) \geq N_k(t)$, and by Wald's identity,

$$\begin{aligned} t &> \mathbb{E}\left[\sum_{m=1}^{\hat{N}_k(t)-1} \hat{D}_{k,m}\right] \geq \mathbb{E}\left[\sum_{m=1}^{\hat{N}_k(t)} \hat{D}_{k,m}\right] - \beta \\ &= \mathbb{E}\left[\hat{N}_k(t)\right] \mathbb{E}\left[\hat{D}_k\right] - \beta \\ &\geq \mathbb{E}[N_k(t)]\left(\delta_k - \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}]\right) - \beta. \end{aligned}$$

Therefore,

$$\mathbb{E}[N_k(t)] \leq \frac{t+\beta}{\delta_k - \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}]},$$

where we used that $\delta_k - \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}] > 0$, which follows from our assumption $\mathbb{P}(D_k \leq \beta) > 0$, finishing the proof of the upper bound. As for the lower bound, by the definition of $N_k(t)$ and Wald's identity we have $\mathbb{E}[N_k(t)]\,\delta_k \geq t$, thus finishing the proof of (5.8). We now prove (5.7), which follows from (5.8) and Wald's second identity given in Lemma 4.1, that is

$$\frac{\mathbb{E}\left[(S_k(t) - N_k(t)\mu)^2\right]}{\mathbb{E}[N_k(t)]^2} = \frac{\sigma_k^2}{\mathbb{E}[N_k(t)]} \geq \frac{\sigma_k^2 \delta_k}{t+\beta} - \frac{\sigma_k^2 \mathbb{E}[D_k \mathbb{I}\{D_k > \beta\}]}{t+\beta}.$$

□

C.3 Proof of Lemma 5.3

Before proceeding with the proof recall the definitions of $(I_m)_{m\geq 1}$, $(T_k(m))_{m\geq 1, k\leq K}$, $(Y_m)_{m\geq 1}$, $(J_m)_{m\geq 1}$ and $(N(t))_{t\geq 0}$: $I_m \in \{1,\dots,K\}$ is the index of the sampler chosen by $\mathcal{A}$ for round m; $T_k(m) = \sum_{s=1}^{m} \mathbb{I}\{I_s = k\}$ is the number of samples obtained from sampler k by the end of round m; $Y_m = X_{I_m, T_{I_m}(m)}$ is the mth sample observed by $\mathcal{A}$; $J_0 = 0$ and $J_m = \sum_{s=1}^{m} D_{I_s, T_{I_s}(s)}$ is the time after $\mathcal{A}$ observes the mth sample, and so the mth round lasts over the time period $[J_{m-1}, J_m)$; and $N(t) = \sum_{m=1}^{\infty} \mathbb{I}\{J_m < t\}$ is the

index of the round at time t (note that $N(t) > 0$ since $t \geq 0$). Thus, $N(t)$ is the number of samples observed after exhausting the time budget t. Note that $\sum_{k=1}^{K} T_k(m) = m$ for any $m \geq 0$ and thus, in particular, $\sum_{k=1}^{K} T_k(N(t)) = N(t)$. Further, let $S_m = \sum_{s=1}^{m} Y_m$ and $S(t) = S_{N(t)}$.

Proof. We proceed similarly to the proof of Lemma 5.2.We first prove (5.11). By the definition of $N(t)$ and since $D_{k,m}$ is nonnegative for all k and m,

$$t \leq J_{N(t)} \leq \sum_{s=1}^{N(t)} D_{I_s, T_{I_s}(s)} = \sum_{k=1}^{K} \sum_{m=1}^{T_k(N(t))} D_{k,m} \,.$$

Notice that $(N(t))_{t\geq 0}$, $(T_k(n))_{n=0,1,\ldots}$ are stopping times w.r.t. the filtration $(\mathcal{F}_m; m \geq 1)$, where $\mathcal{F}_m = \sigma(I_1, J_1, \ldots, I_m, Y_m)$. Therefore, $T_k(N(t))$ is also a stopping time w.r.t. $(\mathcal{F}_m)$. Defining $\overline{T}_k(t) = \mathbb{E}[T_k(N(t))]$, Wald's identity yields

$$t \leq \sum_{k=1}^{K} \mathbb{E}[T_k(N(t))]\, \delta_k = \sum_{k=1}^{K} \overline{T}_k(t) \delta_k \,. \tag{C.1}$$

Then,

$$\mathbb{E}[N(t)] = \sum_{k=1}^{K} \overline{T}_k(t) \geq \frac{1}{\delta_{\max}} \sum_{k=1}^{K} \overline{T}_k(t) \delta_k \geq \frac{t}{\delta_{\max}} \,,$$

finishing the proof of (5.11).

Now, let us turn to showing that (5.10) holds. We first observe

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_{k=1}^{K} S_{k, T_k(N(t))} - T_k(N(t))\mu \right)^2 \right] &= \sum_{k=1}^{K} \mathbb{E}\Big[\big(S_{k, T_k(N(t))} - T_k(N(t))\mu \big)^2 \Big] \\
&\quad + 2 \sum_{k \neq j} \mathbb{E}\Big[\big(S_{k, T_k(N(t))} - T_k(N(t))\mu \big) \big(S_{j, T_j(N(t))} - T_j(N(t))\mu \big) \Big] \\
&= \sum_{k=1}^{K} \overline{T}_k(t) \sigma_k^2,
\end{aligned}$$

where we have used Lemma 4.2 and Wald's second identity (Lemma 4.1). Recall that, by assumption, $\overline{T}_k(t) \leq f(t)$ for all $k \neq k^*$. On the other hand, $\overline{T}_{k^*}(t) \leq \mathbb{E}[N(t)]$ holds trivially. Therefore, using (C.1) and (5.11), and defining $\Delta_{\delta_k} := (\delta_k - \delta_{k^*})^+$ and $\Delta_{\sigma_k^2} := (\sigma_k^2 - \sigma_{k^*}^2)^+$ as well as $\Delta_\sigma := \sum_k \Delta_{\sigma_k^2}$ and $\Delta_\delta := \sum_k \Delta_{\delta_k}$, we obtain

$$
\begin{aligned}
\frac{\mathbb{E}\left[(S(t)-N(t)\mu)^2\right]}{\mathbb{E}[N(t)]^2} &\leq \frac{\sum_{k=1}^K \overline{T}_k(t)\sigma_k^2}{\mathbb{E}[N(t)]^2} \\
&\leq \frac{(\sum_{k=1}^K \overline{T}_k(t)\delta_k)(\sum_{k=1}^K \overline{T}_k(t)\sigma_k^2)}{t\mathbb{E}[N(t)]^2} \\
&= \frac{((\overline{T}_{k^*}(t)\delta_{k^*} + \sum_{k\neq k^*} \overline{T}_k(t)\delta_k)(\overline{T}_{k^*}(t)\sigma_{k^*}^2 + \sum_{k\neq k^*} \overline{T}_k(t)\sigma_k^2)}{t\mathbb{E}[N(t)]^2} \\
&\leq \frac{(\mathbb{E}[N(t)]\,\delta_{k^*} + \sum_{k\neq k^*} \overline{T}_k(t)\Delta_{\delta_k})(\mathbb{E}[N(t)]\,\sigma_{k^*}^2 + \sum_{k\neq k^*} \overline{T}_k(t)\Delta_{\sigma_k^2})}{t\mathbb{E}[N(t)]^2} \\
&\leq \frac{(\mathbb{E}[N(t)]\,\delta_{k^*} + f(t)\Delta_\delta)(\mathbb{E}[N(t)]\,\sigma_{k^*}^2 + f(t)\Delta_\sigma)}{t\mathbb{E}[N(t)]^2} \\
&= \frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{f(t)(\sigma_{k^*}^2\Delta_\delta + \delta_{k^*}\Delta_\sigma)}{t\mathbb{E}[N(t)]} + \frac{f(t)^2\Delta_\delta\Delta_\sigma}{t\mathbb{E}[N(t)]^2} \\
&\leq \frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{f(t)\delta_{\max}(\sigma_{k^*}^2\Delta_\delta + \delta_{k^*}\Delta_\sigma)}{t^2} + \frac{f(t)^2\Delta_\delta\Delta_\sigma\delta_{\max}^2}{t^3} \\
&= \frac{\delta_{k^*}\sigma_{k^*}^2}{t} + \frac{C' f(t)}{t^2} + \frac{C'' f(t)^2}{t^3},
\end{aligned}
$$

where $C' = \delta_{\max}(\sigma_{k^*}^2\Delta_\delta + \delta_{k^*}\Delta_\sigma)$ and $C'' = \Delta_\delta\Delta_\sigma\delta_{\max}^2$. This finishes the proof of the second part of the lemma. □

C.4 KL-Based Confidence Bound on Variance

The following lemma gives a variance estimate based on Kullback-Leibler divergence.

Lemma C.1. *Let $Y_1, \ldots, Y_{2n}$ and $Q_1, \ldots, Q_n$ be two independent sequences of independent and identically distributed random variables taking values in $[0,1]$. Furthermore, for any $t = 1, \ldots, n$, let*

$$
\bar{\mathbb{V}}_{2t} = \frac{1}{2t}\sum_{s=1}^{t} Q_t(Y_{2s} - Y_{2s-1})^2. \tag{C.2}
$$

Then, for any $\delta > 0$,

$$
\mathbb{P}\left(\cup_{t=1}^n \left\{\mathrm{KL}\left(2\bar{\mathbb{V}}_{2t}, 2\mathbb{E}[Q_1]\,\mathbb{V}[Y_1]\right) \geq \frac{\delta}{t}\right\}\right) \leq 2e\lceil\delta\log n\rceil e^{-\delta}\,.
$$

Proof. Equation (5) of Garivier (2013) states that if $Z_1, \ldots, Z_n$ are independent, identically distributed random variables taking values in $[0,1]$ and $\bar{Z}_t = \frac{1}{t}\sum_{s=1}^t Z_s$ for $t = 1, \ldots, n$, then for any $\delta > 0$,

$$
\mathbb{P}\left(\cup_{t=1}^n \left\{\mathrm{KL}\left(\bar{Z}_t, \mathbb{E}[Z_1]\right) \geq \frac{\delta}{t}\right\}\right) \leq 2e\lceil\delta\log n\rceil e^{-\delta}\,.
$$

Defining $Z_t = Q_t(Y_{2t} - Y_{2t-1})^2$, we see that the above conditions on Z_t are satisfied since they are clearly independent and identically distributed for $t = 1, \ldots, n$, and $Z_t \in [0,1]$

since $0 \leq Q_t, Y_{2t-1}, Y_{2t} \leq 1$. Now the statement of the lemma follows since $\mathbb{E}[Z_t] = \mathbb{E}\left[Q_t(Y_{2t} - Y_{2t-1})^2\right] = \mathbb{E}[Q_t]\,\mathbb{E}\left[((Y_{2t} - \mathbb{E}[Y_{2t}]) - (Y_{2t-1} - \mathbb{E}Y_{2t-1}))^2\right] = 2\mathbb{E}[Q_1]\,\mathbb{V}[Y_1]$. □

Corollary C.1. *Let $Y_1, \ldots, Y_{2n}$ and $Q_1, \ldots, Q_n$ be two independent sequences of independent and identically distributed random variables taking values in $[0, 1]$. For any $t = 2, \ldots, n$, let $\bar{\mathbb{V}}_t$ be defined by (C.2) if t is even, and let $\bar{\mathbb{V}}_t = \bar{\mathbb{V}}_{t-1}$ if t is odd. Furthermore, let*

$$\bar{\mathbb{V}}_{t,min} = \inf\{\mu : \mathrm{KL}\left(2\bar{\mathbb{V}}_t, 2\mu\right) \leq \delta/\lfloor t/2 \rfloor\}$$

and

$$\bar{\mathbb{V}}_{t,max} = \sup\{\mu : \mathrm{KL}\left(2\bar{\mathbb{V}}_t, 2\mu\right) \leq \delta/\lfloor t/2 \rfloor\}.$$

Then for any $\delta > 0$, with probability at least $1 - 2e\lceil \delta \log \lfloor n/2 \rfloor \rceil e^{-\delta}$,

$$\max_{2 \leq t \leq n} \bar{\mathbb{V}}_{t,min} \leq \mathbb{E}[Q]\,\mathbb{V}[Y] \leq \min_{2 \leq t \leq n} \bar{\mathbb{V}}_{t,max}\,.$$

Appendix D

Weighted Estimation of a Common Mean

D.1 Proof of Theorem 6.1

Proof. We first establish the mean and variance of μ^*. In particular, for the mean of μ^* we have

$$\mathbb{E}[\mu^*(X_{\mathbf{n}})] = \mathbb{E}\left[\sum_{k=1}^{K} \lambda_k^* \hat{\mu}_k^*(X_{\mathbf{n}})\right] = \sum_{k=1}^{K} \lambda_k^* \mu = \mu\,,$$

where we have used $\sum_{k=1}^{K} \lambda_k^* = 1$. For the variance, we have

$$\mathbb{V}(\hat{\mu}^*(X_{\mathbf{n}})) = \sum_{k=1}^{K} \left(\frac{n_k/\sigma_k^2}{\sum_{j=1}^{K} n_j/\sigma_j^2}\right)^2 \frac{\sigma_k^2}{n_k} = \left(\sum_{j=1}^{K} \frac{n_j}{\sigma_j^2}\right)^{-2} \sum_{k=1}^{K} \frac{n_k}{\sigma_k^2} = \left(\sum_{k=1}^{K} \frac{n_k}{\sigma_k^2}\right)^{-1}. \tag{D.1}$$

To prove that $\hat{\mu}^*$ is UMVU w.r.t. $\mathcal{E}_{\sigma^2}^{\text{Normal}}$ we will use the Cramer-Rao lower bound, which in our setting implies that for any unbiased estimator $\hat{\mu}'$ and any environment $\nu \in \mathcal{E}_{\sigma^2}^{\text{Normal}} \cap \mathcal{E}(\mu)$, when $X_{\mathbf{n}} \sim \nu^{\mathbf{n}}$,

$$\mathbb{V}\big(\hat{\mu}'(X_{\mathbf{n}})\big) \geq \frac{1}{I(\mu)}\,,$$

where $I(\mu)$ is the Fisher information defined by

$$I(\mu) = -\mathbb{E}\left[\frac{\partial^2}{\partial \mu^2} \log(\mathcal{L}(\mu|X_{\mathbf{n}}))\right] = \mathbb{E}\left[\sum_{k=1}^{K}\sum_{t=1}^{n_k} \frac{\partial^2}{\partial \mu^2} \frac{(X_{k,t}-\mu)^2}{2\sigma_k^2}\right] = \sum_{k=1}^{K} \frac{n_k}{\sigma_k^2}\,.$$

This lower bound is constant for all μ and equal to the variance of $\hat{\mu}^*$ implying that $\hat{\mu}^*$ is a UMVU estimator.

To show that $\hat{\mu}^*$ is a minimax estimator for any $\mathcal{E}$ s.t. $\mathcal{E}_{\sigma^2}^{\text{Normal}} \subset \mathcal{E} \subset \mathcal{E}_{\sigma^2}$, we will first derive the Bayes estimator $\hat{\mu}_{\pi_0}$ for the same normally distributed setting above and for some prior $\pi_0(\mu)$ we then construct a sequence of *least favourable* priors $\pi_0, \pi_1, ...$ and show that

$\hat{\mu}_{\pi_0}$ converges to $\hat{\mu}^*$ which allows us to conclude that it is minimax w.r.t. $\mathcal{E}^{\text{Normal}}_{\sigma^2}$. We then extend this result to cover the non-parametric setting.

Specifically, assuming $X_k \sim \mathcal{N}(\mu, \sigma_k)$ and $\pi_0(\mu) = \mathcal{N}(\mu; \mu_0, b^2)$ we can write the (Bayesian) posterior density for μ given the prior π_0 and $x_{\mathbf{n}} \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_K}$ as

$$
\begin{aligned}
p(\mu|x_{\mathbf{n}}) &\propto \pi_0(\mu) \prod_{k=1}^{K} \prod_{i=1}^{n_k} \mathcal{L}(x_{k,i}|\mu) \\
&\propto \exp\left\{\frac{-(\mu_0-\mu)^2}{2b^2}\right\} \prod_{k=1}^{K} \exp\left\{\frac{-n_k(\bar{x}_k-\mu)^2}{2\sigma_k^2}\right\} \\
&= \exp\left\{\frac{-\mu^2}{2}\left(\frac{1}{b^2}+\sum_{k=1}^{K}\frac{n_k}{\sigma_k^2}\right) + \mu\left(\frac{\mu_0}{b^2}+\sum_{k=1}^{K}\frac{n_k\bar{x}_k}{\sigma_k^2}\right) - \left(\frac{\mu_0^2}{2b^2}+\sum_{k=1}^{K}\frac{n_k\bar{x}_k^2}{2\sigma_k^2}\right)\right\} \\
&\propto \mathcal{N}(\mu; \mu_{\pi_0}(x_{\mathbf{n}}), \sigma^2_{\pi_0}(x_{\mathbf{n}})),
\end{aligned}
$$

where $\bar{x}_k := \frac{1}{n_k}\sum_{i=1}^{n_k} x_{k,i}$ and

$$
\mu_{\pi_0}(x_{\mathbf{n}}) := \sigma_n^2\left(\frac{\mu_0}{b^2}+\sum_{k=1}^{K}\frac{n_k\bar{x}_k}{\sigma_k^2}\right), \tag{D.2}
$$

$$
\sigma^2_{\pi_0}(x_{\mathbf{n}}) := \frac{1}{\frac{1}{b^2}+\sum_{k=1}^{K}\frac{n_k}{\sigma_k^2}}. \tag{D.3}
$$

Recall that an estimator $\hat{\mu}$ is said to be *Bayes* with respect to the squared loss, a prior π_0 over the means $\mu \in \mathbb{R}$ and for $\mathcal{E}^{\text{Normal}}_{\sigma^2}$ if it minimizes the average loss $\int L_{\mathbf{n}}(\hat{\mu}; \nu^{\text{Normal}}_{\mu})\pi_0(\mu)d\mu$, where $\nu^{\text{Normal}}_{\mu} = (N(\mu, \sigma_1^2), \ldots, N(\mu, \sigma_K^2))$. It follows from a straightforward application of Fubini's theorem that under the squared loss the model averaged parameter $\hat{\mu}_{\pi_0}(X_{\mathbf{n}}) = \mathbb{E}[\mu|X_{\mathbf{n}}]$ (given by Eq. (D.2)) is a Bayes estimator for π_0 (see Corollary 1.2 Ch. 5 Lehmann and Casella (1998)). Also, the loss of the $\hat{\mu}_{\pi_0}$ estimator is given by

$$
\begin{aligned}
L_{\mathbf{n}}(\hat{\mu}_{\pi_0}; \nu^{\text{Normal}}_{\mu}) &= \mathbb{V}(\hat{\mu}_{\pi_0}(X_{\mathbf{n}})) + \text{bias}(\hat{\mu}_{\pi_0}(X_{\mathbf{n}}))^2 \\
&= \frac{1}{\frac{1}{b^2}+\sum_{k=1}^{K}\frac{n_k}{\sigma_k^2}} + \left(\frac{1}{1+b^2\sum_{k=1}^{K}\frac{n_k}{\sigma_k^2}}(\mu_0-\mu)\right)^2.
\end{aligned}
$$

Now, by considering a sequence of increasingly less informative priors $\pi_1, \pi_2, \ldots$ defined with fixed μ_0 and increasing $b_1^2 < b_2^2 < \ldots$ we can see that as $b_n^2 \to \infty$ we have $\hat{\mu}_{\pi_n} \to \hat{\mu}^*$ and that $L_{\mathbf{n}}(\hat{\mu}_{\pi_n}; \nu^{\text{Normal}}_{\mu}) \uparrow (\sum_{k=1}^{K} n_k/\sigma_k^2)^{-1}$. Since $L_{\mathbf{n}}(\hat{\mu}^*; \nu^{\text{Normal}}_{\mu})$ is the limit of Bayes risks and is constant with respect to μ, clearly $L_{\mathbf{n}}(\hat{\mu}^*; \nu^{\text{Normal}}_{\mu}) = \sup_{\mu} L_{\mathbf{n}}(\hat{\mu}^*; \nu^{\text{Normal}}_{\mu})$ and we therefore can conclude that it is minimax for $\mathcal{E}^{\text{Normal}}_{\sigma^2}$ (see Theorem 1.12 Ch. 5 of Lehmann and Casella (1998)).

Now take any $\mathcal{E}$ such that $\mathcal{E}_{\sigma^2}^{\text{Normal}} \subset \mathcal{E} \subset \mathcal{E}_{\sigma^2}$. Since $\sup_{\nu \in \mathcal{E}_{\sigma^2}^{\text{Normal}}} L_{\mathbf{n}}(\hat{\mu}^*; \nu) = \sup_{\nu \in \mathcal{E}} L_{\mathbf{n}}(\hat{\mu}^*; \nu)$, by Lemma 1.15 Ch. 5 of Lehmann and Casella (1998), $\hat{\mu}^*$ is also minimax optimal for $\mathcal{E}$, concluding the proof.

□

D.2 Proof of Theorem 6.2

Proof. We begin by first decomposing the random loss

$$\hat{L}_n := \Big(\sum_{k=1}^{K} \lambda_k \underbrace{(\hat{\mu}_k - \mu)}_{Z_k} \Big)^2 ,$$

depending on whether ξ holds: $\hat{L}_n = \hat{L}_n \mathbb{I}\{\xi\} + \hat{L}_n \mathbb{I}\{\xi^C\}$. Thanks to $(\hat{\mu}_k - \mu)^2 \leq \bar{b}_k^2 \leq b^2$, $\hat{L}_n \leq b$ and hence

$$\mathbb{E}\Big[\hat{L}_n\Big] \leq \mathbb{E}\Big[\hat{L}_n \mathbb{I}\{\xi\}\Big] + b^2 \, \mathbb{P}\big(\xi^C\big) \ .$$

Thus, it remains to bound the first term in the last expression. By assumption, $\lambda_k^* = \frac{n_k/\sigma_k^2}{\sum_j n_j/\sigma_j^2} > 0$ for all $1 \leq k \leq K$. By expanding the defining expression of $\hat{L}_n$ we see that

$$\hat{L}_n = \sum_{k,j} \lambda_k^* \lambda_j^* Z_k Z_j \frac{\lambda_k \, \lambda_j}{\lambda_k^* \, \lambda_j^*} = \Big(\sum_k \lambda_k^* Z_k \Big)^2 + \sum_{k,j} \lambda_k^* \lambda_j^* Z_k Z_j \left(\frac{\lambda_k \, \lambda_j}{\lambda_k^* \, \lambda_j^*} - 1 \right) . \tag{D.4}$$

Let $\varepsilon_k = \frac{\hat{\sigma}_k^2 + \Delta_k}{\sigma_k^2}$. By algebra, followed by applying Jensen's inequality,

$$\frac{\lambda_k}{\lambda_k^*} = \frac{1}{\varepsilon_k} \left(\frac{\sum_i \frac{n_i}{\hat{\sigma}_i^2 + \Delta_i}}{\sum_i \frac{n_i}{\sigma_i^2}} \right)^{-1} = \frac{1}{\varepsilon_k} \left(\sum_i \lambda_i^* \frac{1}{\varepsilon_i} \right)^{-1} \leq \frac{1}{\varepsilon_k} \left(\sum_i \lambda_i^* \, \varepsilon_i \right) .$$

Note that on $\xi_k = \{|\hat{\sigma}_k^2 - \sigma_k^2| \leq \Delta_k\}$ (hence also on ξ), $1 \leq \varepsilon_k \leq 1 + \frac{2\Delta_k}{\sigma_k^2}$ and by simple algebra also $1/\varepsilon_k \geq 1 - \frac{2\Delta_k}{\sigma_k^2}$. Thus, on ξ,

$$1 - \frac{2\Delta_k}{\sigma_k^2} \leq \frac{\lambda_k}{\lambda_k^*} \leq 1 + 2 \sum_i \lambda_i^* \, \frac{\Delta_i}{\sigma_i^2} =: 1 + 2\zeta \, . \tag{D.5}$$

Depending on whether $k = j$ we use different techniques to bound the terms appearing in the second sum of the last expression of (D.4). In particular, when $k = j$,

$$\mathbb{E}\left[Z_k^2 \left(\left(\frac{\lambda_k}{\lambda_k^*} \right)^2 - 1 \right) \mathbb{I}\{\xi\} \right] \leq 4\zeta(1+\zeta) \, \mathbb{E}\big[Z_k^2\big] = 4\zeta(1+\zeta) \, \frac{\sigma_k^2}{n_k}$$

and

$$\mathbb{E}\left[\mathbb{I}\{\xi\} \sum_k (\lambda_k^*)^2 Z_k^2 \left(\left(\frac{\lambda_k}{\lambda_k^*} \right)^2 - 1 \right) \right] \leq \frac{4\zeta(1+\zeta)}{\sum_k \frac{n_k}{\sigma_k^2}} \, .$$

When $k \neq j$, we first write $\mathbb{E}\Big[Z_k Z_j \left(\frac{\lambda_k}{\lambda_k^*}\frac{\lambda_j}{\lambda_j^*} - 1\right)\mathbb{I}\{\xi\}\Big] = \mathbb{E}\Big[Z_k Z_j \frac{\lambda_k}{\lambda_k^*}\frac{\lambda_j}{\lambda_j^*}\mathbb{I}\{\xi\}\Big] - \mathbb{E}[Z_k Z_j \mathbb{I}\{\xi\}]$. Now, by independence,

$$\mathbb{E}\left[Z_k Z_j \frac{\lambda_k}{\lambda_k^*}\frac{\lambda_j}{\lambda_j^*}\mathbb{I}\{\xi\}\right] = \mathbb{E}\left[\prod_{i\neq k,j}\mathbb{I}\{\xi_i\}\right]\mathbb{E}\left[Z_k\frac{\lambda_k}{\lambda_k^*}\mathbb{I}\{\xi_k\}\right]\mathbb{E}\left[Z_j\frac{\lambda_j}{\lambda_j^*}\mathbb{I}\{\xi_j\}\right].$$

Further, $-\mathbb{E}[Z_k Z_j \mathbb{I}\{\xi\}] = -\mathbb{E}\big[Z_k Z_j (1 - \mathbb{I}\{\xi^C\})\big] = \mathbb{E}\big[Z_k Z_j \mathbb{I}\{\xi^C\}\big] \leq \bar{b}_k \bar{b}_j \mathbb{P}\big(\xi^C\big)$. Hence, to bound the cross term it remains to bound $|\mathbb{E}\Big[Z_k \frac{\lambda_k}{\lambda_k^*}\mathbb{I}\{\xi_k\}\Big]|$. For this, we have

$$\begin{aligned}|\mathbb{E}\Big[Z_k \frac{\lambda_k}{\lambda_k^*}\mathbb{I}\{\xi_k\}\Big]| &= \Big|\mathbb{E}\Big[Z_k\left(\frac{\lambda_k}{\lambda_k^*} - \frac{m_k + M_k}{2}\right)\mathbb{I}\{\xi_k\}\Big] + \frac{m_k + M_k}{2}\,\mathbb{E}\big[Z_k(1 - \mathbb{I}\{\xi_k^C\})\big]\Big| \\ &\leq \frac{M_k - m_k}{2}\,\mathbb{E}[|Z_k|] + \frac{m_k + M_k}{2}\,\bar{b}_k\,\mathbb{P}\big(\xi_k^C\big)\,,\end{aligned}$$

where m_k (and M_k) are deterministic constants that almost surely lower (respectively, upper) bound $\frac{\lambda_k}{\lambda_k^*}$ over ξ_k. Thus, for the cross-term, via an application of Cauchy-Schwartz to upper bound $\mathbb{E}[|Z_k|]$ by $\sqrt{\mathbb{E}[|Z_k|^2]} = \frac{\sigma_k}{\sqrt{n_k}}$, we have

$$\begin{aligned}\mathbb{E}\Big[Z_k Z_j \left(\frac{\lambda_k}{\lambda_k^*}\frac{\lambda_j}{\lambda_j^*} - 1\right)\mathbb{I}\{\xi\}\Big] &\leq \bar{b}_k\bar{b}_j\mathbb{P}\big(\xi^C\big) + \\ &\left\{\frac{M_k - m_k}{2}\,\frac{\sigma_k}{\sqrt{n_k}} + \frac{M_k + m_k}{2}\,\bar{b}_k\,\mathbb{P}\big(\xi_k^C\big)\right\}\left\{\frac{M_j - m_j}{2}\,\frac{\sigma_j}{\sqrt{n_j}} + \frac{M_j + m_j}{2}\,\bar{b}_j\,\mathbb{P}\big(\xi_j^C\big)\right\}.\end{aligned}$$

and so

$$\begin{aligned}\sum_{k\neq j}\lambda_k^*\lambda_j^*\mathbb{E}\Big[Z_k Z_j\left(\frac{\lambda_k}{\lambda_k^*}\frac{\lambda_j}{\lambda_j^*} - 1\right)\mathbb{I}\{\xi\}\Big] &\leq \Big(\sum_k \lambda_k^*\bar{b}_k\Big)^2\mathbb{P}\big(\xi^C\big) \\ &+ \Big(\sum_k \lambda_k^*\Big\{\frac{M_k - m_k}{2}\,\frac{\sigma_k}{\sqrt{n_k}} + \frac{M_k + m_k}{2}\,\bar{b}_k\,\mathbb{P}\big(\xi_k^C\big)\Big\}\Big)^2.\end{aligned}$$

By (D.5), we can choose $m_k = 1 - \frac{2\Delta_k}{\sigma_k^2}$ and $M_k = 1 + 2\zeta$. Hence, $\frac{M_k - m_k}{2} = \zeta + \frac{\Delta_k}{\sigma_k^2} = O(n^{-1/2})$ and $\frac{m_k + M_k}{2} = 1 + \zeta - \frac{\Delta_k}{\sigma_k^2} = 1 + O(n^{-1/2})$. Putting together everything we get

$$\begin{aligned}R_n(\hat{\mu}_{\text{UCB-W}}) &= n^2\left\{\mathbb{E}\Big[\hat{L}_n\Big] - L_n^*\right\} \\ &\leq n^2\left\{2b^2\,\mathbb{P}\big(\xi^C\big) + \frac{4\zeta(1+\zeta)}{\sum_k \frac{n_k}{\sigma_k^2}} + \Big(\sum_k \lambda_k^*\frac{(M_k - m_k)\sigma_k}{2\sqrt{n_k}} + \frac{m_k + M_K}{2}\,\bar{b}_k\,\mathbb{P}\big(\xi_k^C\big)\Big)^2\right\},\end{aligned}$$

where we used that $\mathbb{E}\big[(\sum_k \lambda_k^* Z_k)^2\big] = L_n^*$. We see that here the leading term in the inside bracket is

$$4\frac{\zeta}{n\sum_k \frac{\rho_k}{\sigma_k^2}} = 10\sqrt{2}n^{-3/2}\,\frac{\sum_i \lambda_i^*\frac{\bar{b}_i\sqrt{\log(4K/\delta)}}{\sigma_i^2\sqrt{\rho_i}}}{\sum_k \frac{\rho_k}{\sigma_k^2}} = 10\sqrt{2}n^{-3/2}\,\frac{\sum_i \frac{\bar{b}_i\sqrt{\rho_i\log(4K/\delta)}}{\sigma_i^4}}{\Big(\sum_k \frac{\rho_k}{\sigma_k^2}\Big)^2},$$

thus, letting $C = 10\sqrt{2}\left(\sum_k \frac{\rho_k}{\sigma_k^2}\right)^{-2}$ and using $(a+b)^2 \leq 2(a^2 + b^2)$, algebra recovers the main result. □

Remark D.1. *An alternative argument uses Hölder's inequality and avoids the boundedness condition as follows: For any* $\varepsilon > 0$, $\mathbb{E}\Big[\hat{L}_n \mathbb{I}\{\xi^C\}\Big] \leq \mathbb{E}\Big[\hat{L}_n^{1+\varepsilon}\Big]^{\frac{1}{1+\varepsilon}} \mathbb{E}\big[\mathbb{I}\{\xi^C\}\big]^{\frac{\varepsilon}{1+\varepsilon}}$. *Now, by Jensen's inequality,* $\hat{L}_n^p \leq \sum_{k=1}^K \lambda_k |Z_k|^{2(1+\varepsilon)} \leq \max_k |Z_k|^{2(1+\varepsilon)}$. *Hence,* $\mathbb{E}\Big[\hat{L}_n \mathbb{I}\{\xi^C\}\Big] \leq \mathbb{E}\big[\max_k |Z_k|^{2(1+\varepsilon)}\big]^{\frac{1}{1+\varepsilon}} \mathbb{P}\big(\xi^C\big)^{\frac{\varepsilon}{1+\varepsilon}}$. *Since we expect* $Z_k \sim N(0, \sigma_k^2/n_k)$ *(asymptotically) and they are independent, we expect* $\mathbb{E}\big[\max_k |Z_k|^{2(1+\varepsilon)}\big]$ *to be well-controlled and in particular scale with* $\log(K)$.

D.3 Proof of Theorem 6.3

Proof. For convenience we denote $\Pi_k := \xi_k \cap \mathcal{N}_k$ and $\Pi := \bigcap_{k=1}^K \Pi_k$ and we decompose the random loss as

$$\hat{L}_n := \Big(\sum_{k=1}^K \lambda_k \underbrace{(\hat{\mu}_k - \mu)}_{Z_k}\Big)^2,$$

depending on whether Π holds: $\hat{L}_n = \hat{L}_n \mathbb{I}\{\Pi\} + \hat{L}_n \mathbb{I}\{\Pi^C\}$. As above, this gives

$$\mathbb{E}\Big[\hat{L}_n\Big] \leq \mathbb{E}\Big[\hat{L}_n \mathbb{I}\{\Pi\}\Big] + b^2\, \mathbb{P}\big(\Pi^C\big)\ .$$

We will address the latter term through assumptions on ξ and $\mathcal{N}$ and focus on bounding the first term. Specifically we have that

$$\hat{L}_n = \sum_{k,j} \lambda_k^* \lambda_j^* Z_k Z_j \frac{\lambda_k\, \lambda_j}{\lambda_k^*\, \lambda_j^*} = \Big(\sum_k \lambda_k^* Z_k\Big)^2 + \sum_{k,j} \lambda_k^* \lambda_j^* Z_k Z_j \left(\frac{\lambda_k\, \lambda_j}{\lambda_k^*\, \lambda_j^*} - 1\right). \tag{D.6}$$

In what follows we will often need to bound quantities depending on the number of pulls (on Π) so for convenience we define $\bar{N}_k := \mathbb{E}[N_k] + \Delta_{N_k}$, $\underline{N}_k := \mathbb{E}[N_k] - \Delta_{N_k}$, and $\lambda_j^* \leq \bar{\lambda}_j^* := \frac{\bar{N}_j}{\sigma_j^2} / \sum_k \frac{\underline{N}_k}{\sigma_k^2}$. Defining $\varepsilon_k = \frac{\hat{\sigma}_k^2 + \Delta_{\sigma_k^2}(N_k)}{\sigma_k^2}$ as well, we then have

$$\frac{\lambda_k}{\lambda_k^*} = \frac{1}{\varepsilon_k} \left(\frac{\sum_i \frac{N_i}{\hat{\sigma_i}^2 + \Delta_{\sigma_i^2}(N_k)}}{\sum_i \frac{N_i}{\sigma_i^2}}\right)^{-1} = \frac{1}{\varepsilon_k} \left(\sum_i \lambda_i^* \frac{1}{\varepsilon_i}\right)^{-1} \leq \frac{1}{\varepsilon_k} \left(\sum_i \lambda_i^*\, \varepsilon_i\right).$$

On Π_k we have that $\{|\hat{\sigma}_k^2 - \sigma_k^2| \leq \Delta_{\sigma_k^2}(N_k) \leq \Delta_{\sigma_k^2}(\underline{N}_k)\}$ (hence also on Π), $1 \leq \varepsilon_k \leq 1 + \frac{2\Delta_{\sigma_k^2}(N_k)}{\sigma_k^2}$ and $1/\varepsilon_k \geq 1 - \frac{2\Delta_{\sigma_k^2}(\underline{N}_k)}{\sigma_k^2}$. Thus, on Π,

$$1 - \frac{2\Delta_{\sigma_k^2}(\underline{N}_k)}{\sigma_k^2} \leq \frac{\lambda_k}{\lambda_k^*} \leq 1 + 2\sum_i \bar{\lambda}_i^* \frac{\Delta_{\sigma_i^2}(\underline{N}_k)}{\sigma_i^2} =: 1 + 2\zeta\,. \tag{D.7}$$

Depending on whether $k = j$ we use different techniques to bound the expectation of the terms appearing in the second sum of the last expression of Eq. (D.6). In particular, when

$k = j$,

$$\mathbb{E}\left[Z_k^2\left(\left(\frac{\lambda_k}{\lambda_k^*}\right)^2 - 1\right)\mathbb{I}\{\Pi\}\right] \leq 4\zeta(1+\zeta)\,\mathbb{E}\left[Z_k^2\right].$$

As before, for the cases where $j \neq k$ we first observe $\mathbb{E}\left[Z_k Z_j\left(\frac{\lambda_k\,\lambda_j}{\lambda_k^*\,\lambda_j^*} - 1\right)\mathbb{I}\{\Pi\}\right] = \mathbb{E}\left[Z_k Z_j \frac{\lambda_k\,\lambda_j}{\lambda_k^*\,\lambda_j^*}\mathbb{I}\{\Pi\}\right] - \mathbb{E}[Z_k Z_j \mathbb{I}\{\Pi\}] \leq \mathbb{E}\left[Z_k Z_j \frac{\lambda_k\,\lambda_j}{\lambda_k^*\,\lambda_j^*}\mathbb{I}\{\Pi\}\right] + \bar{b}_k\bar{b}_j\mathbb{P}\big(\Pi^C\big)$. We can upper bound the first term here as before

$$\begin{aligned}
\mathbb{E}\left[Z_k Z_j \frac{\lambda_k\,\lambda_j}{\lambda_k^*\,\lambda_j^*}\mathbb{I}\{\Pi\}\right] &\leq |\mathbb{E}\left[Z_k \frac{\lambda_k\lambda_j}{\lambda_k^*\lambda_j^*}\mathbb{I}\{\Pi\}\right]| \\
&\leq \left|\mathbb{E}\left[Z_k\left(\frac{\lambda_k\lambda_j}{\lambda_k^*\lambda_j^*} - \frac{m_{k,j}+M_{k,j}}{2}\right)\mathbb{I}\{\Pi\}\right]\right. \\
&\quad \left. + \frac{m_{k,j}+M_{k,j}}{2}\,\mathbb{E}\big[Z_k Z_j(1-\mathbb{I}\{\Pi^C\})\big]\right| \\
&\leq \frac{M_{k,j}-m_{k,j}}{2}\,\mathbb{E}[|Z_k Z_j|] + \frac{m_{k,j}+M_{k,j}}{2}\,\bar{b}_k\bar{b}_j\,\mathbb{P}\big(\Pi^C\big) \\
&\leq \frac{M_{k,j}-m_{k,j}}{2}\sqrt{\mathbb{E}\big[Z_k^2\big]\,\mathbb{E}\left[Z_j^2\right]} + \frac{m_{k,j}+M_{k,j}}{2}\,\bar{b}_k\bar{b}_j\,\mathbb{P}\big(\Pi^C\big)\ ,
\end{aligned}$$

where $m_{k,j}$ (and $M_{k,j}$) are deterministic constants that almost surely lower (respectively, upper) bound $\frac{\lambda_k\lambda_j}{\lambda_k^*\lambda_j^*}$ over Π and the final inequality follows from Cauchy-Schwartz. Here we can choose $m_{k,j} = (1 - \frac{2\Delta_k(N_k)}{\sigma_k^2})(1 - \frac{2\Delta_j(N_j)}{\sigma_j^2})$ and $M_{k,j} = (1+2\zeta)^2$. Hence, by assumption we have that $\sum_k \mathbb{E}[N_k] \leq C_f t$, which implies that $\frac{M_{k,j}-m_{k,j}}{2} \leq 2\zeta + 2\zeta^2 + \frac{\Delta_k(N_k)}{\sigma_k^2} + \frac{\Delta_j(N_j)}{\sigma_j^2} = O(t^{-1/2})$ and $\frac{m_k+M_k}{2} = 1 + 2\zeta + 2\zeta^2 + 2\frac{\Delta_k(N_k)\Delta_j(N_j)}{\sigma_k^2\sigma_j^2} = 1 + O(t^{-1/2})$. Putting this all together we arrive at

$$\begin{aligned}
\mathbb{E}[L(\hat{\mu}) - L(\hat{\mu}^*)] &\leq \mathbb{E}\left[\sum_{k,j}\lambda_k^*\lambda_j^* Z_k Z_j\left(\frac{\lambda_k\lambda_j}{\lambda_k^*\lambda_j^*} - 1\right)\right] \\
&\leq \sum_{k,j}\bar{\lambda}_k^*\bar{\lambda}_j^*\mathbb{E}\left[Z_k Z_j\left(\frac{\lambda_k\lambda_j}{\lambda_k^*\lambda_j^*} - 1\right)\right] \\
&\leq \sum_k(\bar{\lambda}_k^*)^2 4\zeta(1+\zeta)\,\mathbb{E}\big[Z_k^2\big] \\
&\quad + \sum_{k\neq j}\bar{\lambda}_k\bar{\lambda}_j\left(\frac{C'}{\sqrt{t}}\sqrt{\mathbb{E}\big[Z_k^2\big]\,\mathbb{E}\left[Z_j^2\right]} + (1+\frac{C''}{\sqrt{t}})\bar{b}_k\bar{b}_j\,\mathbb{P}\big(\Pi^C\big)\right)\ ,
\end{aligned}$$

where we observe that $\mathbb{E}\big[Z_k^2\big] \leq \frac{\sigma_k^2}{\mathbb{E}[N_k]}\left(1 + \frac{2c}{\sqrt{\mathbb{E}[N_k]}}\right)^2 + f(b,\delta)$ by Lemma 5.1 with $f(b,\delta)$ capturing the higher-order terms. □

D.4 Concentration Inequalities

In this section we give some concentration inequalities that we need in the text.

D.4.1 The Hoeffding-Azuma Inequality

Lemma D.1 (Hoeffding-Azuma Inequality)**.** *Let $(X_t)_{1\le t\le n}$ be a martingale difference process such that $|X_t| \le r_t$ a.s. with some $(r_t)_{1\le t\le n}$ deterministic constants. Then, for any $c \ge 0$,*

$$\mathbb{P}\left(\sum_{t=1}^{n} X_t \ge c\right) \le \exp\left(-\frac{c^2}{2\sum_{t=1}^{n} r_t^2}\right),$$
$$\mathbb{P}\left(\sum_{t=1}^{n} X_t \le -c\right) \le \exp\left(-\frac{c^2}{2\sum_{t=1}^{n} r_t^2}\right).$$

Inverting this inequality gives the following:

Corollary D.1. *Under the same conditions as in Lemma D.1 on (X_t), for any $0 \le \delta \le 1$,*

$$\mathbb{P}\left(\sum_{t=1}^{n} X_t \le \sqrt{2\left(\sum_{t=1}^{n} r_t^2\right)\log\left(\frac{1}{\delta}\right)}\right) > 1-\delta,$$
$$\mathbb{P}\left(\sum_{t=1}^{n} X_t \ge -\sqrt{2\left(\sum_{t=1}^{n} r_t^2\right)\log\left(\frac{1}{\delta}\right)}\right) > 1-\delta.$$

D.4.2 Concentration of the Sample Variance

The following inequality can be extracted from the proof of Theorem 10 of Maurer and Pontil (2009):

Lemma D.2. *Let $n \ge 2$, $(X_t)_{1\le t\le n}$ be independent, $[0,1]$-valued random variables and define*

$$V_n = \frac{1}{n(n-1)} \sum_{1\le i<j\le n} (X_i - X_j)^2$$

to be the sample variance of $(X_t)_t$, and let $\sigma_n^2 = \mathbb{E}[V_n]$ be the expected sample variance. Then, for any $0 \le \delta \le 1$,

$$\mathbb{P}\left(\sigma_n^2 \le V_n + \sigma_n\sqrt{\frac{2\log(1/\delta)}{n-1}}\right) > 1-\delta,$$
$$\mathbb{P}\left(\sigma_n^2 \ge V_n - \sigma_n\sqrt{\frac{2\log(1/\delta)}{n-1}} - \frac{\log(1/\delta)}{n-1}\right) > 1-\delta.$$

Note that the sample variance as defined here is the same as the value $\frac{1}{n-1}\sum_{t=1}^{n}(X_t - \frac{1}{n}\sum_{s=1}^{n} X_s)^2$, a form that is perhaps more commonly used to define the sample variance. Note that if $(X_t)_t$ are identically distributed then $\sigma_n^2 = \sigma^2 := \mathbb{V}(X_t)$, the common variance of the random variables $(X_t)_t$. o While the above result gives $1-\delta$ confidence upper

and lower bounds on σ_n^2, these depend on σ_n^2, hence cannot be used directly in the lack of knowledge of σ_n^2. A simple way to get fully empirical confidence bounds is to upper bound σ_n. For example, when $(X_t)_t$ is i.i.d., $\mathbb{E}[V_n] = \mathbb{V}(X_t) \le 1/4$ where the upper bound is Popoviciu's inequality for the variances. (More generally, if $(X_t)_t$ is i.i.d., but they belong to $[a, a+r]$, then $\mathbb{E}[V_n] \le r^2/4$). This leads to the following result:

Corollary D.2. *Let $n \ge 2$, $(X_t)_{1\le t\le n}$ be i.i.d., $[0,1]$-valued random variables with common variance σ^2. Then, for any $0 \le \delta \le 1$,*

$$\mathbb{P}\left(\sigma^2 \le V_n + \sqrt{\frac{\log(1/\delta)}{2(n-1)}}\right) > 1-\delta\,,$$
$$\mathbb{P}\left(\sigma^2 \ge V_n - \sqrt{\frac{\log(1/\delta)}{2(n-1)}} - \frac{\log(1/\delta)}{n-1}\right) > 1-\delta\,.$$

An alternative way to estimate the common variance of a sequence of i.i.d. random variables is to use $V_n' = \frac{1}{n}\sum_{t=1}^{\lfloor n/2\rfloor}(X_{2t-1}-X_{2t})^2$. It is not hard to see that indeed $\mathbb{E}[V_n'] = \sigma^2$. The advantage of using V_n' is that it is quite simple to derive confidence bounds for σ^2 based on V_n'. For example, Hoeffding's inequality can be used directly to obtain lower and upper confidence bounds for σ^2: For any $0 \le \delta \le 1$, for n even,

$$\mathbb{P}\left(\sigma^2 \le V_n' + \sqrt{\frac{\log(1/\delta)}{n}}\right) > 1-\delta \quad \text{and} \quad \mathbb{P}\left(\sigma^2 \ge V_n' - \sqrt{\frac{\log(1/\delta)}{n}}\right) > 1-\delta\,.$$

Comparing with Corollary D.2, we see that this approach loses a factor of $\sqrt{2}$ in the length of the confidence interval. By considering normally distributed random variables, it is not hard to see that this is indeed the price that one pays for the simpler analysis.

One can use Lemma D.2 to get tighter fully empirical confidence bounds for σ_n^2. In this approach, to derive (say) an upper bound on σ_n^2, one finds the maximum value of σ^2 such that $\sigma^2 \le V_n + \sigma\sqrt{\frac{2\log(1/\delta)}{n-1}}$ holds. Similarly, for a lower bound on σ_n^2, one finds the smallest *nonnegative* value of σ^2 such that $\sigma^2 \ge V_n - \sigma\sqrt{\frac{2\log(1/\delta)}{n-1}} - \frac{\log(1/\delta)}{n-1}$ holds (the lower bound is zero when $V_n \le \frac{\log(1/\delta)}{n-1}$). Solving these inequalities gives the following result:

Corollary D.3. *Under the conditions of Lemma D.2, for any $0 \le \delta \le 1$, with probability at least $1-\delta$,*

$$\begin{aligned}\sigma_n^2 &\le V_n + \sqrt{\frac{2\log(1/\delta)}{n-1}\left(V_n + \frac{\log(1/\delta)}{2(n-1)}\right)} + \frac{\log(1/\delta)}{n-1} \\ &\le V_n + \sqrt{V_n\frac{2\log(1/\delta)}{n-1}} + 2\,\frac{\log(1/\delta)}{n-1}\end{aligned}$$

holds. Similarly, for any $0 \leq \delta \leq 1$*, with probability at least* $1 - \delta$*,*

$$\begin{aligned}\sigma_n^2 &\geq \left(V_n - \sqrt{\frac{2\log(1/\delta)}{n-1}\left(V_n - \frac{\log(1/\delta)}{2(n-1)}\right)} \right)_+ \\ &\geq \left(V_n - \sqrt{V_n \frac{2\log(1/\delta)}{n-1}} - \frac{\log(1/\delta)}{n-1} \right)_+ .\end{aligned}$$

Interestingly, the obtained bounds on σ_n^2 take the same exact form as Lemma D.2 except that σ_n^2 is replaced by V_n, in the upper bound an additional $O(1/n)$ term appears, while and the coefficients of the $O(1/n)$ terms are slightly increased.