Essays in simulation and stochastic processes

Dan MacKinlay

A dissertation submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

Supervised by Dr Zdravko Botev

School of Mathematics and Statistics
Faculty of Science
UNSW Sydney

2020

PLEASE TYPE

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: MacKinlay

First name: Daniel Other name/s: Bruce

Abbreviation for degree as given in the University calendar: PhD

School: Mathematics and Statistics Faculty: Science

Title: Essays in stochastic processes

Abstract 350 words maximum: (PLEASE TYPE)

This thesis is concerned with two topics rooted in the analysis of time-series.
In the first, we improve the estimation of rare-event probabilities by stochastic simulation.
The proposed method, quasi-monotone splitting, uses generalized splitting to estimate integrals with respect to intractable target distributions by instead estimating them with respect to the terminal state of certain Markov chains, allowing us to use time series methods to study them.
We employ two innovations to this end:
Problem constraints are exploited to derive a simple, efficient estimation strategy automatically for a tractable problem class. The performance of the estimator is then improved through the use of survival analysis and extreme value theory, in which near-optimal parameters can be derived with minimal intervention.
We demonstrate applications of this algorithm to a variety of wireless reliability problems.
The performance of the resulting algorithms are competitive with specialized Monte Carlo estimators for specific problems, and provide novel estimators for problems previously lacking known, efficient estimators.
The second topic is audio signal analysis.
An important task here is style transfer which attempts to synthesize a new signal from two others, a source and a target. The new synthetic signal should possess the microscopic “stylistic” statistics of the source, and the macroscopic “semantic” statistics of the target.
We solve this problem using mosaicing, which decomposes the source signal into microscopic sub-samples, superimposing them to produce the new synthetic signal whose macroscopic statistics approximate the target.
In such models, one chooses parameters by minimising some loss function which ideally approximates acoustic similarity as perceived by a human listener.
We leverage the insight that human pitch perception is related to the local autocorrelogram of a signal to construct a novel loss function based on a difference between autocorrelograms.
This, in combination with a signal approximation method based on orthogonal matching pursuits, results in a novel synthesis algorithm called autocorrelogram mosaicing.
This algorithm is the only one we know of with public code that can mosaic with arbitrary pitch transposition of source audio, enabling style transfer between differently tuned instruments while maintaining musical consonance.
The strength and weakness of this algorithm for various source materials is demonstrated.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed 

Date 19/10/20

COPYRIGHT STATEMENT

Signed 

Date 19/6/2020

AUTHENTICITY STATEMENT

Signed 

Date 19/6/2020

Table of Contents

List of Tables

List of Figures

Abstract

This thesis is concerned with two topics rooted in the analysis of time-series.

The first is in the area of rare-event simulation. This is of particularly important in areas where rare but crucial events are of great interest, such as finance and wireless networking. Many such rare event probabilities are numerically intractable to calculate, even where all the parameters are known. Such quantities must be estimated by approximate means. One such approximate means is stochastic simulation, which has a number of desirable properties, such as small or zero bias, and well-understood asymptotic behaviour.

We improve these stochastic estimators for certain special classes of problems by a new technique. Our *quasi-monotone splitting* uses *generalized splitting* (Botev et al. 2012) to estimate rare event probabilities by transforming them into estimators with respect to some "nice" Markov chain. This allows us to treat the target quantities as a time series estimation problem, using all the power of time series methods. The performance of the resulting algorithms are competitive with specialized Monte Carlo estimators for specific problems, and provide novel estimators for problems previously lacking known, efficient estimators.

The second topic pertains to a different times series: audio signals. An important task here is *style transfer*, which attempts to synthesize a new *mosaic* signal from two others, a *source* and a *target*. Intuitively, the mosaic signal should have the 'content' of the target, but the 'style' of the source. This problem is important in industrial applications such as voice and music synthesis. Our approach to this problem uses *mosaicing* style transfer. This method decomposes the source signal into very short snippets, then superimposes those snippets into a a new mosaic with the large-scale structure matching the target. There are many mosaicing-type methods; ours is unique because it is based on the autocorrelogram, which characterises signals by their self-similarity, which is a simple approximation to hu-

man pitch perception. Autocorrelograms are usually considered computationally intractable, but we handle them using a combination of a sparse basis decomposition and a stochastic simulation method to find good local matches. The result is a novel synthesis algorithm called *autocorrelogram mosaicing*. We apply it to a number of musically relevant tasks and compare it with other benchmark tasks. Although quantitative measures are difficult in this domain, the result is qualitatively effective and produces novel and aesthetically interesting effects.

Acknowledgements

As every graduate student knows, the facade of solitary endeavour in a doctorate is a flimsy one behind which many labour. The assertion that I have made here of originality is true in the pure and narrow legal sense that my fingers were upon the keyboard that generated this text. But insofar as a thesis is more than a typing exercise, every line of novel prose and legally-distinct intellectual property which I arrogate for myself I have purchased in the currency of other's efforts on my behalf.

The theoretical frameworks of chapter 2 of this thesis derives from research conducted in the course of writing the paper Ben Rached et al. (2020), and I amd indebted to my co-authors Nadhir Ben Rached and Zdravko Botev for the many fruitful discussions and even more fruitful disagreements

Part II of this thesis is a work published in a conference paper (MacKinlay and Botev 2019) at the Congress of the International Society For Music Information Retrieval, of which this author was primary author. It has been lightly edited for stylistic consistency and for some minor corrections.

I am indebted to the sponsorship of people who have by their sweat and love gifted me the time, space, insight and energy to create this document for which they can receive only the faint credit in this least-read of sections in this least-read of documents. Thus, recognising the faintness of the praise: to the very limit permitted by my statement of originality, I hereby acknowledge the vital contribution of certain people for any good in this thesis.

Thank you, Miriam Lyons, Emma Vine, Stella Kirkby, Owen Brasier, Cris Baldwin, Ray Burns, Ed Stewart and Alice Adamson who have cooked for me, washed for me, and tolerated my emotional absence and tardiness with the chores in order to carve out a few extra moments of writing time.

Thank you Susannah Waters, Haya Aldosari, Kam Hung Yau, Thomas Scheckter, Fiona Kim and many other colleagues have made the grad student office, when it has been open, a convivial and excellent place. Special thanks to Bruce Henry, who has been an excellent head of school as the university has dismantled itself around us.

I am indebted to my kind proof-readers Miriam Lyons (again), Maria Findeisen,

Tom Morris, Ed Stewart, James Nichols, Samantha Wilson, and the true endurance effort of Rob Salomone. Useful technical discussion with Edwin Bonilla, Nadhir Ben Rached, James Nichols (again), Rob Salomone (again), Boris Beranger and my supervisor Zdravko Botev have made a material difference to the progress of my ideas. Nadhir Ben Rached and Zdravko Botev have in addition been the co-authors and researchers on the paper for which we developed the material in chapter 2.

Thank you also to my father, Alistair MacKinlay, and Heidi Emery, who helped me survive my Master's thesis, which was how I got into this mess, and lent me their granny flat to weather the start of my thesis confinement.

Finally, my most most effusive and least timely thanks must go to my mother, Roslyn MacKinlay née Clarke, the first female engineer to graduate in Western Australia, and knitter of my hot water bottle cover, who asked me to get around to finishing this project before she died. Thanks for everything. I'm sorry I was six months late.

Nomenclature

Nomenclature

$\bar{F}$	$\bar{F} = 1 - F$, most often used in the complementary cumulative distribution function.
$\mathscr{S}$	Algebras are written using curly font $\mathscr{S}$.
$\mathbb{N}^0$	$\mathbb{N} \cup \{0\}$
$\mathbb{E}$	Expectation
$\mathbb{I}$	Indicator function
$\lim_{\varepsilon \searrow x} f(\varepsilon)$	Limit from above of $f(\varepsilon)$ as $\varepsilon \to x, \varepsilon > x$
$\mathrm{Unif}(\mathcal{X})$	If $\mathcal{X}$ is finite, $\mathrm{Unif}(\mathcal{X})$ is the discrete uniform distribution over the set $\mathcal{X}$. If $\mathcal{X}$ is an interval then it is the continuous uniform distribution over that interval.
$\mathbb{P}$	Probability
$X \overset{\mathrm{D}}{=} Y$	X is identically distributed to Y
$X \perp\!\!\!\perp Y \mid Z$	X is independent of Y given Z
$X \sim F$	The law of X is given by F, where F may be a measure, or the CDF associated to a measure
$\mathcal{S}$	Sets are written using calligraphic font $\mathcal{S}$.
$x \overset{\mathrm{def}}{=} y$	x is defined to be y
CCDF	Complementary Cumulative Distribution Function

CDF	Cumulative Distribution Function
CMC	Crude Monte Carlo
DFT	Discrete Fourier Transform
ENRV	Effort Normalized Relative Variance
IID	Independent, Identically Distributed
IS	Importance Sampling
MC	Monte Carlo
MCMC	Markov Chain Monte Carlo
MFCC	Mel Frequency Cepstral Coefficients
PSD	Power Spectral Density
QMS	Quasi-monotone Splitting
RE	Relative Error
RV	Random Variable
STFT	Short Time Fourier Transform
WNRV	Work Normalized Relative Variance

Chapter 1

Introduction

1.1 A thesis in two parts

This thesis is concerned with two topics which over the course of time have evolved divergently.

Much like the dinosaur and the chicken, the family relationship between these is no longer obvious. For our current purposes we present these parts separately, each comparatively independent.

Nonetheless, they have indeed speciated from a single common ancestor. That common ancestor is the theory of time-indexed stochastic processes, and it is our hope that in the future their shared history will reveal strange and new relations between them.

In the first part, we consider the use of stochastic processes in Monte Carlo estimation. Specifically, we improve the estimation of rare-event probabilities by stochastic simulation using Lévy processes. The proposed method, *quasi-monotone splitting* uses *generalized splitting* (Botev et al. 2012) to estimate integrals with respect to intractable target distributions by instead estimating them with respect to the terminal state of a certain Markov chain. This allows us to use time series methods to study these processes.

We employ two innovations to this end:

1. Problem constraints are exploited to derive a simple, efficient estimation strategy automatically for a tractable problem class, and

2. The performance of the estimator is improved through the use of survival analysis and extreme value theory, in which near-optimal parameters can be derived with minimal intervention.

We demonstrate applications of this algorithm to a variety of wireless reliability problems. The performance of the resulting algorithms are competitive with specialized Monte Carlo estimators for specific problems, and provide novel estimators for problems previously lacking known, efficient estimators. Some of the methods in this section were developed for a paper with several co-authors which has now been published (Ben Rached et al. 2020).

The second part covers audio signal analysis. An important task here is *style transfer*, which attempts to synthesize a new signal from two others, a *source* and a *target*. The new synthetic signal should possess the microscopic "stylistic" statistics of the source, and the macroscopic "semantic" statistics of the target. We solve this problem using *mosaicing* style transfer, which decomposes the source signal into microscopic snippets, superimposing them to produce the new synthetic signal whose macroscopic statistics approximate the target.

In such models, one chooses parameters by minimising some loss function which ideally approximates acoustic similarity as perceived by a human listener. We leverage the insight that human pitch perception is related to the local autocorrelogram of a signal to construct a novel loss function based on a difference between autocorrelograms. This, in combination with a signal approximation method based on orthogonal matching pursuits, results in a novel synthesis algorithm called *autocorrelogram mosaicing*. This algorithm is the only one we know of with public code that can mosaic with arbitrary pitch transposition of source audio, enabling style transfer between differently tuned instruments while maintaining musical consonance. The strength and weakness of this algorithm for various source materials is demonstrated.

1.2 Probability background

We introduce our choice of terminological and notational conventions, in probability theory in particular, that will be useful throughout, and essential for the first

part of the thesis in particular.

In this thesis, most of the random objects we encounter are random variables taking values in state space $\mathbb{R}^d$. A random variable X has associated law $\mu(A) = \mathbb{P}[A]$ on the state space for an X-measurable event A. We treat X through its cumulative distribution function (CDF) on $\mathbb{R}^d$, which contains the information we need to characterise the law (or distribution) of the random variable. The CDF of X is a function $F : \mathbb{R}^d \to [0, 1])$ such that if $\mathsf{X} \sim \mu$ then $F(\boldsymbol{x}) = \mathbb{P}(\mathsf{X} \in [-\infty, x_1] \times \cdots \times [-\infty, x_d])$, for $\boldsymbol{x} = [x_1\ x_2\ \ldots\ x_d]$. i.e., $F(\boldsymbol{x}) = \mu([-\infty, x_1] \times \cdots \times [-\infty, x_d])$. It is thus right continuous and non-decreasing in each coordinate of its argument. When F is the CDF of a continuous random variable (i.e., one with no atoms) it is convenient to discuss the density $f(\boldsymbol{x}) = \mathrm{d}F(\boldsymbol{x})/\mathrm{d}\boldsymbol{x}$. This is not always feasible; we consider, for example, distributions of discrete support which do not possess density functions with respect to the Lebesgue measure on real line. We allow integrals with respect to functions of bounded variation in the Riemann-Stieltjes sense, in particular, with respect to CDFs. For the $\mathbb{R}^d$-valued random variables we consider, all the following expectations coincide, where they are defined:

$$\begin{aligned}\mathbb{E}\phi(\mathsf{X}) &= \int \phi(\boldsymbol{x})\mu(\mathrm{d}\boldsymbol{x}) \text{ as a Lebesgue-Stieltjes integral} \\ &= \int \phi(\boldsymbol{x})\mathrm{d}F(\boldsymbol{x}) \text{ as a Riemann-Stieltjes integral} \\ &= \int \phi(\boldsymbol{x})f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \text{ as a Riemann integral.}\end{aligned}$$

In this context we may meaningfully describe the distribution of X though a CDF, $\mathsf{X} \sim F$ or through a law $\mathsf{X} \sim \mu$ and mean the same thing, which is that X is a random variable with law μ, which induces CDF F over the state space. We have collapsed iterated integrals over a vector valued integrand into a single integral sign to avoid a tedium of multiple integral signs. Each of these is interpreted as an iterated integral over the coordinates. For example,

$$\int \phi(\boldsymbol{x})f(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int \cdots \int \phi([x_1\ \cdots\ x_d]^\top)\mathrm{d}x_1 \ldots \mathrm{d}x_d.$$

Many of the random objects we consider are what we would refer to as stochastic process, i.e., families of random variables indexed by some set $\sqcup_{\max} \subseteq \mathbb{R}$. We em-

phasise that a random object is a stochastic process by writing it as $\{\mathsf{X}(t)\}_{t\in\sqcup_{\max}}$.

We have been using a notational convention from machine learning literature — e.g. Domke (2020) — that when discussing a random variable X we use a sans serif font. This has the convenient property that we can assume capital serif letters to represent a CDF and lower-case serif letters to represent a density *per default*, so that $\mathsf{X} \sim F$ is not ambiguous.

$\{\mathsf{G}\} \sim F_1 \times F_2 \times \dots F_d$ denotes that the coordinates of the state of random process $\{\mathsf{G}\} = (\mathsf{G}_1, \mathsf{G}_2, \dots, \mathsf{G}_d)$ are mutually independent with laws $\{\mathsf{G}_i\} \sim F_i, i = 1, 2, \dots, d$. In this case we understand it to mean that the joint distribution of the coordinates of X is given by a product measure $\mu_1 \times \dots \times \mu_d$, where μ_i is the measure associated to CDF F_i. In the common case of d independent components with identical laws $F = F_1 = F_2 = \dots = F_d$ we write $\{\mathsf{G}\} \sim F^{\times d}$. We allow this notation also for independent stochastic processes sharing a common time index.

We write variables in boldface $\boldsymbol{x}$ rather than normal weight x to emphasise that they are $\mathbb{R}^d$ vectors rather than scalars. Likewise, we boldface random variables who take vector values. We have already used the convention that a boldfaced vector's are per default written as non-boldfaced versions of the sign denoting that vector, so that $\boldsymbol{x} = [x_1\, x_2\, \dots\, x_d]$ and x_i is understood to be a coordinate of $\boldsymbol{x}$.

A hatted symbol $\hat{\theta}$ denotes an estimator of some estimand, $\theta\in\Theta$. Target estimands are mostly finite positive numbers so $\Theta = (0, \infty)$, although some other estimands are important. For example, we can estimate entire CDFs. An estimator is a statistic, which is to say, a function of some set of random observed or simulated data $\mathcal{D}\in\Upsilon$ such that $\hat{\theta} : \Upsilon\to\Theta$. Although estimators are themselves random variables, we do not write them in sans serif fonts. Usually the dependence on the data $\mathcal{D}$ is suppressed and we write $\hat{\theta}$ rather than $\hat{\theta}(\mathcal{D})$.

Part I

Quasi-monotone splitting

In this part we introduce the machinery of splitting-based rare event estimation, the kind of problems that it is meant to solve, and the alternative methods by which it can be solved. The context and importance are introduced first, in Section 2.1. Subsequently, we develop the technical definitions and alternative methods that provide the technical landscape in which our method is set.

Our own contribution, the *quasi-monotone* method, was developed for a published paper (Ben Rached et al. 2020), to which the author is a major contributor. The presentation of these results, in Chapter 3 is based upon that work but includes a revised presentation and some additional results. In Chapter 4 we extend this work substantially with new, previously-unpublished results. These further analyse and improve upon the efficiency of the quasi-monotone method by allowing it to self-tune its free parameters.

Implementations of all methods, in the open-source MATLAB-like language Julia, are available at https://github.com/danmackinlay/MonotoneSampling., made freely available for extension, and further research.

Chapter 2

Rare event estimation

We introduce the background to the field of rare event simulation, including the structure and difficulties of the problem of estimating quantities related to rare events. We explain several alternative estimators for these quantities, as well as measures of efficiency of these estimators.

2.1 Background

Rare event problems arise in many application areas where events of small likelihood are of great importance in some behaviour of interest in a system.

Examples are common in actuarial and financial risk models. Ruin probabilities for a firm paying our contracts at some random rate are of great interest (McNeil, Frey, and Embrechts 2005) and are by design intended to be small and by necessity must be quantified. Under non-trivial assumptions these probabilities do not have analytic forms and must be estimated. Similarly, in transmission systems such as power or communications networks, overall system reliability requires good estimation of probabilities of rare, but potentially extremely costly failures (Botev, L'Ecuyer, and Tuffin 2018; Botev et al. 2012) and indeed one of the early applications of the Generalised splitting method (upon which we build) is in network reliability estimation.

For the current purposes we draw examples particularly from the wireless reliability field. In this domain, network outages are modelled by probability dis-

tributions over signal and noise power levels. Different combinations of signals of interest, other interfering signals, and of background noise, are all represented by different combinations of random variables, and an outage in signal transmission is corresponds to a rare tail event. Within such a model, maintaining a specified level of wireless network reliability requires controlling the rate of outages represented by these by these rare tail events (Simon and Alouini 2005). Under different assumptions upon the transmission infrastructure and noise distributions we may find a rich variety of different rare event models.

More generally, we might wish to estimate not only the magnitude of the probability of a certain rare event, but also an integral conditional upon it; for example, we might wish to know the magnitude of our shortfall in a financial ruin context. Our method can also be bent to this task.

2.2 Method

We are concerned with problems of estimating certain expectations of a random variable or process $\phi(\mathsf{X}) \sim F$ with values restricted to a target set, $\mathcal{L}$. There are two problems of interest to us in this context. Firstly, the restricted expectation,[1]

$$\mathbb{E}_F\left[\phi(\mathsf{X})\mathbb{I}\{\mathcal{L}\}\right] = \int_{\mathcal{L}} \phi(\boldsymbol{x})\mathrm{d}F(\boldsymbol{x}). \tag{2.1}$$

Secondly, the closely-related conditional expectation,

$$\theta = \mathbb{E}_F\left[\phi(\mathsf{X}) \mid \mathcal{L}\right] = \frac{\int_{\mathcal{L}} \phi(\boldsymbol{x})\mathrm{d}F(\boldsymbol{x})}{\mathbb{P}_F[\mathcal{L}]}. \tag{2.2}$$

Here $\phi : \mathbb{R}^d \to \mathbb{R}$, X is an $\mathbb{R}^d$-valued random vector, and $\mathcal{L}$ is some *target event*. For our purposes, we consider only $\theta > 0$. Hereafter, where the distribution is clear we suppress the subscript, writing $\mathbb{P}$ for $\mathbb{P}_F$ and $\mathbb{E}$ for $\mathbb{E}_F$.

Specifically, we propose methods which are effective with the *rare* event setting

[1] The reader, having read the theoretical introduction section, will recall that this integral with respect to F is to be interpreted as a Riemann-Stieltjes integral with respect to the CDF F which, for the $\mathbb{R}^d$-valued random variables we consider, coincides with the Lebesgue-Stieltjes integral with respect to the law $\mathsf{X} \sim \mu$ of the random variable, where $\int_{\mathcal{L}} \phi(\boldsymbol{x})\mu(\mathrm{d}\boldsymbol{x}) = \int_{\mathcal{L}} \phi \mathrm{d}F(\boldsymbol{x})$ with $F(\boldsymbol{x}) = \mu([-\infty, x_1] \times \cdots \times [-\infty, x_d])$.

— that is, we want to guarantee that our methods are effective for low-probability target events where $\mathbb{P}[\mathcal{L}] = \mathbb{E}[\mathbb{I}\{\mathcal{L}\}] \ll 1$, say $\mathbb{P}[\mathcal{L}] = 10^{-6}$.

More generally, we wish to analyse the performance of methods across a family of similar problems where the target rare event is parameterized by some scalar $\varepsilon > 0$. We write the parameterized target rare event $\mathcal{L}_\varepsilon$ and the associated quantities θ_ε and ℓ_ε.

Definition 2.1 (Rarity parameter)**.** We call an event parameter $\varepsilon \in \mathbb{R}$ a *rarity parameter* if $\mathbb{P}[\mathcal{L}_\varepsilon]$ is monotone non-increasing in ε and $\lim_{\varepsilon \searrow 0} \mathbb{P}[\mathcal{L}_\varepsilon] = 0$.

Where we are not discussing behaviour of the estimator with respect to a rarity parameter, we suppress it.

Example 2.1. A case of (2.1) of particular importance is the *flat* case where $\phi \equiv 1$ and the quantity of interest is the probability of the rare event itself, i.e.,

$$\ell \stackrel{\text{def}}{=} \mathbb{E}\left[\mathbb{I}\{\mathcal{L}\}\right] = \mathbb{P}[\mathcal{L}]. \tag{2.3}$$

We reserve the symbol ℓ for estimands of this type.

Example 2.2. An illustrative case of a problem of the rare-event conditional expectation form (2.2) is the *conditional excess* over threshold κ. Here, $\mathcal{L}_\kappa \stackrel{\text{def}}{=} \{S(\boldsymbol{X}) > \kappa\}$ and we wish to estimate

$$\theta(\kappa) = \mathbb{E}\left[S(\boldsymbol{X}) \mid S(\boldsymbol{X}) > \kappa\right] = \frac{\mathbb{E}\left[S(\boldsymbol{X})\mathbb{I}\{S(\boldsymbol{X}) > \kappa\}\right]}{\mathbb{P}[S(\boldsymbol{X}) > \kappa]} \tag{2.4}$$

for some *importance function* $S : \mathbb{R}^d \to \mathbb{R}$.[2] The conditional excess is needed in important risk models, being used, for example, in the *expected shortfall* measure for portfolios of risky assets (McNeil, Frey, and Embrechts 2005). It also occurs in our own calculations in the extreme-value tail approximations of Section 4.1. Observe that, taking $S : \boldsymbol{x} \mapsto \sum_i x_i$ where all the components are non-negative, $\varepsilon = 1/\kappa$ becomes a rarity parameter for this problem.

[2]The name of this function indicates a relationship to 'vanilla' importance sampling, with the distinction that it is a degenerate case of such a function, taking values only in $\{0, 1\}$.

2.3 Rare event estimation via Monte Carlo

Often, the expectation of interest is not analytically tractable and thus we must estimate it numerically. If we can simulate according to the distribution[3] F, obtaining an approximation to the quantity of interest via stochastic simulation may provide a tractable alternative. This is the *Monte Carlo* (MC) approach. Stochastic approximation of such rare event-restricted integrals is our major topic throughout Part I. Here, we give a brief outline of Monte Carlo theory required for this purpose. In-depth treatments may be found in standard Monte Carlo monographs such as Asmussen and Glynn (2007, p. VI), Kroese, Taimre, and Botev (2011, Ch 10), Rubinstein and Kroese (2016, Ch 9), Bucklew (2004), and Rubino and Tuffin (2009).

To begin, we introduce the *Crude Monte Carlo* (CMC) estimator. The generic CMC method for integrals of the form

$$\theta = \mathbb{E}\left[\zeta(X)\right] = \int \zeta(\boldsymbol{x}) \mathrm{d}F(\boldsymbol{x}) \tag{2.5}$$

approximates the value of the integral by replacing the CDF F with an empirically estimated CDF. The empirical CDF is constructed from from simulations $\mathcal{D}_n \overset{\text{def}}{=} \{X_{(1)}, \ldots, X_{(n)}\}$, drawing each $X_{(i)} \sim F$ independently. Then, the *crude empirical CDF estimate* is

$$\hat{F}^{\text{CMC}}(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{X_{(i)} \leq \boldsymbol{x}\}. \tag{2.6}$$

Here "$\leq$" is taken coordinate-wise, i.e. $\boldsymbol{y} \leq \boldsymbol{x} \Leftrightarrow (y_1 \leq x_1) \cap (y_2 \leq x_2) \cap \cdots \cap (y_d \leq x_d)$. This is the empirical CDF associated with the empirical measure

$$\hat{\mu}^{\text{CMC}}(A) = = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{X_{(i)} \in A\} \tag{2.7}$$

for (Borel-measurable) $A \subset \mathbb{R}^d$. All the empirical distributions we obtain from simulation have a similar form, although we change the generating mechanism for

[3] We recall that, as mentioned in the theoretical section p. 2, we treat the laws of random variables through their CDFs.

the variates $X_{(i)}$ and/or weight the samples in the various Monte Carlo estimators we use.

Substituting $\hat{F}^{\mathrm{CMC}}$ for F in (2.5) turns the expectation, interpreted as a Riemann-Stieltjes integral, into

$$\hat{\theta}^{\mathrm{CMC}} = \int \zeta(\boldsymbol{x}) \mathrm{d}\hat{F}^{\mathrm{CMC}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \zeta(X_{(i)}). \tag{2.8}$$

We say an estimator is *unbiased* if $\mathbb{E}\hat{\theta} = \theta$. A CMC estimator is unbiased provided the integral in question is finite, since

$$\mathbb{E}\hat{\theta} = \mathbb{E}\frac{1}{n}\sum_{i=1}^{n} \zeta(X_{(i)}) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\zeta(X_{(i)}) = \frac{1}{n} n \mathbb{E}\zeta(X_{(1)}) = \mathbb{E}\zeta(X_{(1)}) = \theta. \tag{2.9}$$

An important class of problems is the estimation of the tail probability.

Example 2.3. Where $\phi \equiv 1$ (2.3) we obtain $\zeta : \boldsymbol{x} = \mathbb{I}\{\boldsymbol{x} \in \mathcal{L}\}$ and the CMC estimator (2.8) becomes

$$\hat{\ell}^{\mathrm{CMC}}(n) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{X_{(i)} \in \mathcal{L}\}. \tag{2.10}$$

The accuracy of CMC, like most MC methods, depends upon the amount of computational effort invested to obtain the estimand, which often scales with some parameter like n. We quantify the computational effort through the (possibly random) amount of time dedicated to running the algorithm under specified conditions. We parameterize MC estimator run time via an *effort parameter* η. When needed, we make the dependence on η explicit for a given estimator by writing it as an argument, $\hat{\theta}(\eta)$. Effort parameters are chosen so that a greater effort parameter implies a greater wall-clock execution time. We return to this point momentarily.

An important property of MC estimators is *consistency*: the estimator converges in probability to the estimand as $\eta \to \infty$. That is,

$$\lim_{\eta\to\infty} \mathbb{P}[|\hat{\theta}(\eta) - \theta| > \delta] = 0 \tag{2.11}$$

for any $\delta > 0$. In the case of the CMC estimator, a natural effort parameter is $\eta = n$, the number of independent and identically distributed (IID) realizations we generate to construct our estimate. For finite $\zeta(\mathsf{X})$ it follows immediately from the strong law of large numbers that $\hat{\theta}^{\mathrm{CMC}} \xrightarrow{\text{a.s.}} \theta$ in n, which implied (2.11), so the Crude Monte Carlo estimator (2.8) is consistent.

To quantify how close a Monte Carlo estimator is "on average" to its corresponding estimand, we examine its variance, denoted $\mathrm{Var}[\hat{\theta}]$. We often report this in terms of the standard error $\mathrm{se}(\hat{\theta}) \stackrel{\text{def}}{=} \sqrt{\mathrm{Var}[\hat{\theta}]}$.

Example 2.4 (Variance of the CMC estimator)**.** We note that as a sum of independent Bernoulli trials, the distribution of the CMC estimator for (2.3) is $(n\hat{\ell}^{\mathrm{CMC}}(n)) \sim \mathrm{Binom}(n, \ell)$. It follows that $\mathbb{E}\hat{\ell}^{\mathrm{CMC}}(n) = \ell$ and

$$\mathrm{Var}[\hat{\ell}^{\mathrm{CMC}}(n)] = \ell(1-\ell)/n, \tag{2.12}$$

thus

$$\mathrm{se}[\hat{\ell}^{\mathrm{CMC}}(n)] = \sqrt{\frac{\ell(1-\ell)}{n}}. \tag{2.13}$$

In rare event problems, standard error is not necessarily a useful property. We are generally interested in the behaviour of the error relative to the magnitude of the estimand. For strictly positive estimands (our focus throughout), it is reasonable to analyze efficiency in terms of *relative error*.

Definition 2.2 (Relative error)**.** The *relative error* of an estimator $\hat{\theta}$ of estimand $\theta > 0$ is given by

$$\mathrm{re}(\hat{\theta}) \stackrel{\text{def}}{=} \frac{\mathrm{se}(\hat{\theta})}{\theta} = \frac{\sqrt{\mathrm{Var}[\hat{\theta}]}}{\theta}. \tag{2.14}$$

Example 2.5 (Relative error of the CMC estimator)**.** When estimating the tail probability (2.3), with $\ell \ll 1$ we can calculate the relative error of our CMC

probability estimate as

$$\mathrm{re}(\hat{\ell}^{\mathrm{CMC}}(n)) = \frac{\sqrt{\ell(1-\ell)/n}}{\ell} \tag{2.15}$$

$$= \sqrt{\frac{1-\ell}{\ell n}} \tag{2.16}$$

$$\approx \frac{1}{\sqrt{n\ell}} \quad \text{as } 1-\ell \approx \ell \text{ for small } \ell. \tag{2.17}$$

Relative error is the quantity we aim to control even in the small-probability limit as $\ell \to 0$. There are many metrics which can be used to quantify the scaling of relative error with rarity. Overviews are available in, for example, Cancela, Rubino, and Tuffin (2005) and L'Ecuyer et al. (2010). A desirable guarantee of small-probability error is given by *bounded relative error* (BRE) (L'Ecuyer et al. 2010; Shahabuddin 1994). An estimator $\hat{\theta}_\varepsilon$ is said to possess this quality with respect to rarity parameter ε and effort parameter η if

$$\limsup_{\varepsilon \to 0} \mathrm{re}^2(\hat{\theta}_\varepsilon(\eta)) < K(\eta) \tag{2.18}$$

for some finite $K(\eta)$ which does not depend upon ε. Some of the state-of-the-art estimators we compare our own estimators with are known to possess BRE. The CMC estimator (2.17) fails to possess BRE, since we can always make the relative error arbitrarily large by decreasing ε and hence ℓ_ε.

Many problems of importance have no known estimators for which BRE can be established. For such problems, we usually satisfy ourselves with weaker guarantees, the most important of which is *logarithmic efficiency* (Asmussen and Rubinstein 1995), now widely used (e.g. Rubinstein and Kroese 2016, p. 389). An estimator which achieves logarithmic efficiency has the property that the expected computational cost of attaining a given relative error grows at a polynomial rate in $\log(\ell)$ (Kriman and Rubinstein 1995). A sufficient condition for an unbiased estimator $\hat{\theta}$ of θ with rarity parameter ε to be logarithmically efficient is (Asmussen and Rubinstein 1995)

$$\lim_{\varepsilon \to 0} \frac{\log \mathbb{E}\left[\hat{\theta}_\varepsilon^2\right]}{\log \theta_\varepsilon} = 2. \tag{2.19}$$

Example 2.6 (CMC and logarithmic efficiency)**.** The CMC estimator is not logarithmically efficient since

$$\lim_{\varepsilon\to 0} \frac{\log \mathbb{E}\left[(\hat{\ell}_\varepsilon^{\mathrm{CMC}})^2\right]}{\log \ell_\varepsilon} = \lim_{\varepsilon\to 0} \frac{\log\left(\mathrm{Var}[\hat{\ell}_\varepsilon^{\mathrm{CMC}}] + \mathbb{E}[\hat{\ell}_\varepsilon^{\mathrm{CMC}}]^2\right)}{\log \ell_\varepsilon} \tag{2.20}$$

$$= \lim_{\varepsilon\to 0} \frac{\log\left(\ell_\varepsilon(1-\ell_\varepsilon) + \ell_\varepsilon^2\right)}{\log \ell_\varepsilon} \tag{2.21}$$

$$= \lim_{\varepsilon\to 0} \frac{\log \ell_\varepsilon}{\log \ell_\varepsilon} \tag{2.22}$$

$$= 1 < 2. \tag{2.23}$$

Here we make the significance and interpretation of the effort parameters concrete. The wall-clock time (implicitly, with respect to a given configuration of hardware and software) to compute $\hat{\theta}(\eta)$ for a given run of the estimator with a specified effort parameter η is denoted $\mathrm{time}(\hat{\theta}(\eta))$. We use this wall-clock run-time to quantify how much *work* a given algorithm requires. We naturally prefer estimators which can attain a certain accuracy with the least possible amount of work. The run-time can be random, in which case we quantify work with the *expected* (wall-clock) time $C(\hat{\theta}(\eta) \stackrel{\mathrm{def}}{=} \mathbb{E}[\mathrm{time}(\hat{\theta}(\eta))]$. Henceforth, we make the approximation (which holds reasonably well for all estimators considered herein) that, for a given estimator $\hat{\theta}(\eta)$, $C(\hat{\theta}(\eta))$ is approximately linear in some appropriately-chosen η. That is, there is a constant C_0 such that $C(\hat{\theta}(\eta)) \approx C_0\eta$. These run times by assumption have small variance relative to their expectation. The effort parameter η usually scales proportionally with the expected number of random realizations required in some method.

To quantify the scaling of estimator error with estimator effort, we use effort- or work-normalization of the variance. We define the *work-normalized variance* (WNV) as

$$\mathrm{WNV}(\hat{\theta}(\eta)) \stackrel{\mathrm{def}}{=} \mathrm{Var}[\hat{\theta}(\eta)]C(\hat{\theta}(\eta)). \tag{2.24}$$

Sometimes it is convenient to use instead the *effort*-normalized relative variance,

$$\mathrm{ENV}(\hat{\theta}(\eta)) \stackrel{\mathrm{def}}{=} \mathrm{Var}[\hat{\theta}(\eta)]\eta. \tag{2.25}$$

In the case, as stipulated here, that $C(\hat{\theta}(\eta)) \approx C_0\eta$, these efficiency measures are related by

$$\mathrm{WNV}(\hat{\theta}(\eta)) \approx C_0\,\mathrm{ENV}(\hat{\theta}(\eta)). \tag{2.26}$$

For rare-event problems we translate WNV and ENV into relative measures by substituting *relative* error in place of absolute error.

Definition 2.3 (Effort normalized relative variance)**.** The *effort normalized relative variance* (ENRV) of an estimator $\theta(\eta)$ with effort parameter η is

$$\mathrm{ENRV}(\hat{\theta}(\eta)) \stackrel{\mathrm{def}}{=} \frac{\mathrm{ENV}(\hat{\theta})}{\theta^2} = \frac{\mathrm{Var}[\hat{\theta}(\eta)]\eta}{\theta^2} = \mathrm{re}^2(\hat{\theta}(\eta))\eta. \tag{2.27}$$

Definition 2.4 (Work normalized relative variance)**.** With all terms as given in ENRV, the Work Normalized Relative Variance (WNRV) is

$$\mathrm{WNRV}(\hat{\theta}) \stackrel{\mathrm{def}}{=} \frac{\mathrm{Var}[\hat{\theta}(\eta))C(\hat{\theta}(\eta)]}{\theta^2} = \mathrm{re}^2(\hat{\theta}(\eta))C(\hat{\theta}(\eta)). \tag{2.28}$$

This form for work-normalization is widely used in the analysis of Monte Carlo estimators due to an argument of Glynn and Whitt (1992) which states that, over a broad class of estimators, the large-effort asymptotic relation

$$\lim_{\eta\to\infty} \mathrm{WNV}(\hat{\theta}(\eta)) = \text{constant} \tag{2.29}$$

holds in probability. In the rare-event setting we use the equivalent relation

$$\lim_{\eta\to\infty} \mathrm{WNRV}(\hat{\theta}_\varepsilon(\eta)) = \text{constant}(\varepsilon). \tag{2.30}$$

Estimators which, in the large-effort asymptotic limit, attain constant WNRV are said to scale at the *canonical rate*.

In the large-effort asymptotic limit, at least, there is a natural ranking of the accuracy of estimators which accounts for both computational time and the variance of the estimator — an estimator with a lower asymptotic WNRV asymptotically outperforms one with a higher asymptotic WNRV. This large-effort asymptotic argument motivates the use of canonical-rate variance normalization as a rule-of-thumb metric to compare Monte Carlo estimators. Small effort performance

guarantees do not necessarily follow from the large-effort asymptotic guarantees, although this nicety is often assumed away, and we assume the normalization is reasonable even in small effort regimes.

Effort- and work- normalized performance are closely related, but have different strengths. We can compare estimators using effort normalization where we know *a priori* that this implies a valid comparison of wall-clock execution time. The virtue of this approach is that we need not keep identical computer hardware at hand to re-run simulations for comparison. When we are comparing estimation methods with incommensurable effort parameters, for example because the estimators are structurally different, or differently parameterized, we use work normalization. This situation can arise if, say, an alternative estimator $\hat{\theta}'(\eta)$ of the same quantity simulates different random variates to accomplish the same estimation to some baseline $\hat{\theta}(\eta)$.

Example 2.7 (Efficiency of CMC)**.** The CMC estimator has a constant ENRV/WNRV with respect to a fixed target event, since, by (2.12)

$$\mathrm{ENRV}(\hat{\ell}(n)) = \frac{\mathrm{Var}[\hat{\ell}(n)]}{\ell^2} n = \frac{(1-\ell)}{\ell}. \tag{2.31}$$

Since by assumption $C^{\mathrm{CMC}}(\hat{\ell}(n)) = C_0^{\mathrm{CMC}} n$ for some constant C_0^{CMC} we have

$$\mathrm{WNRV}(\hat{\ell}(n)) = C_0^{\mathrm{CMC}} \frac{(1-\ell)}{\ell}. \tag{2.32}$$

Clearly neither of these depends upon the effort parameter, n.

We estimate these quantities for given estimators empirically, as statistics of an ensemble of $R > 1$ independent replications of the estimator in question. Usually we use the natural sample estimator for a given quantity, e.g. taking sample variance as a true estimate of the estimator variance. The exception is in calculating expected wall-clock time $C(\hat{\theta})$ and derived quantities. Empirical measurements of wall-clock time produce noisy estimates of actual computational effort, being dependent upon details of the hardware and software, and sensitive to interference from other demands upon the system. We address the former problem by ensuring that we hold hardware fixed and use, as far as possible, comparable software and

execution conditions. We address the second problem with robust statistics: the expected execution time $C(\hat{\theta}(\eta))$ is estimated not by sample mean of the replicates but by α-trimmed sample mean. The trimmed mean reduces the impact of a small number of outlier execution times. We write $\text{mean}(\boldsymbol{x}; \alpha)$ to denote *the α-trimmed mean of $\boldsymbol{x}$*. We fix $\alpha = 5\%$ throughout. Writing $\hat{\theta}_{(r)}$ to indicate a particular estimate of $\hat{\theta}$ calculated from a distinct batch of R realizations, we define the following empirical estimators:

$$\hat{\mathbb{E}}(\hat{\theta}) \stackrel{\text{def}}{=} \frac{1}{R} \sum_r \hat{\theta}_{(r)} \tag{2.33}$$

$$\widehat{\text{Var}}(\hat{\theta}) \stackrel{\text{def}}{=} \frac{1}{R} \sum_r \left(\hat{\theta}_{(r)} - \hat{\mathbb{E}}(\hat{\theta}) \right)^2 \tag{2.34}$$

$$\hat{\text{re}}(\hat{\theta}) \stackrel{\text{def}}{=} \frac{\sqrt{\widehat{\text{Var}}[\hat{\theta}]}}{\hat{\mathbb{E}}(\hat{\theta})} \tag{2.35}$$

$$\hat{C}(\hat{\theta}) \stackrel{\text{def}}{=} \text{mean}(\{\text{time}(\hat{\theta}_{(r)}), r = 1, \ldots, R\}; 0.05) \tag{2.36}$$

$$\widehat{\text{ENRV}}(\hat{\theta}(\eta)) \stackrel{\text{def}}{=} \frac{\widehat{\text{Var}}(\hat{\theta})\eta}{\hat{\mathbb{E}}(\hat{\theta})^2} \tag{2.37}$$

$$\widehat{\text{WNRV}}(\hat{\theta}) \stackrel{\text{def}}{=} \frac{\widehat{\text{Var}}(\hat{\theta})\hat{C}(\hat{\theta})}{\hat{\mathbb{E}}(\hat{\theta})^2}. \tag{2.38}$$

We may be concerned in turn about the distribution of these estimators of the Monte Carlo estimator distribution (i.e. the estimator distribution estimator distribution). We estimate confidence intervals for these values using non-parametric bootstrap samples (DiCiccio and Efron 1996; Efron 1979) over the estimator replicates.

In the sequel we compare state-of-the-art Monte Carlo methods for various problems in terms of the efficiency measures discussed. The quasi-monotone splitting method Chapter 3, which is our main contribution, is from a family of methods called *splitting methods*. The benchmark methods for problems we consider fall mostly into the categories of Gibbs sampling Section 2.4, a Markov Chain Monte Carlo (MCMC) method, or *Importance Sampling* (IS) (Section 2.5). We briefly introduce all these methods here.

2.4 Gibbs sampling

Markov Chain Monte Carlo (MCMC) methods form a large subclass of MC methods. They are useful in the case where simulating independently from the target distribution F is intractable, but where it is possible to simulate the paths of a Markov chain whose stationary distribution is F — hence *Markov Chain Monte Carlo*. We focus on one particular method MCMC method, the Gibbs sampler (Geman and Geman 1984), which is convenient in rare-event problems (e.g. Gudmundsson and Hult 2014). In a Gibbs sampler, each coordinate of a vector random sample is updated separately, conditional on fixed values of the other coordinates. A Gibbs sampler can be well-suited to estimating $\mathbb{E}[\phi(\mathsf{X}) \mid \mathcal{L}]$ where the tail event is sufficiently "nice". This is frequently possible in tail events defined by scalar importance functions, (2.51), which are the only type of event we consider here. Gibbs methods are simple, and require no tuning in the basic case. However, they cannot give us estimates of the probability $\mathbb{P}[\mathcal{L}]$.[4] Protracted discussions and extensive references are available in Kroese, Taimre, and Botev (2011) and Rubinstein and Kroese (2016).

For our purposes, introducing the most basic Gibbs sampler suffices to illustrate the principle. We discuss what Kroese, Taimre, and Botev (2011) refer to as the "random sweep" Gibbs sampler, which is constructed as follows: Suppose that $\mathsf{X} \sim F$ is an $\mathbb{R}^d$-valued random variable such that simulating from the joint law $F(x_1, x_2, \ldots, x_d)$ is difficult, but that simulating according to the coordinate-wise conditional law

$$F_{[k]}(\cdot \mid \boldsymbol{x}) \stackrel{\text{def}}{=} F(\cdot \mid \mathsf{X}_1{=}x_1, \ldots, \mathsf{X}_{k-1}{=}x_{k-1}, \mathsf{X}_{k+1}{=}x_{k+1}, \ldots, \mathsf{X}_d{=}x_d) \tag{2.39}$$

is feasible for each k and all $\boldsymbol{x} \in \operatorname{supp}(\mathsf{X})$. The Gibbs sampler chooses each new sample $\mathsf{X}^{(m)} \mid \mathsf{X}^{(m-1)} = \boldsymbol{x}^{(m-1)}$ from the previous realization by updating a single coordinate. At each step m we choose a random $k \in \{1, \ldots, D\}$. The next sample is the same as $\boldsymbol{x}^{(m-1)}$, except that we update the kth component with a sample from the corresponding conditional distribution, $x_k^{(m)} \leftarrow x \sim F_{[k]}$. Suppose that $\mathsf{X}^{(m-1)} \sim F$. Holding k fixed and drawing $\mathsf{X}^{(m)} \sim F_{[k]}(\cdot \mid \mathsf{X}^{(m-1)})$, the resulting

[4] See Gudmundsson and Hult (2014) for an example where additional restrictions on the form of the problem enable estimating the $\mathbb{P}[\mathcal{L}]$.

transition kernel samples from F also. This fact follows directly from the definition of conditional probability. Writing this for compactness as a calculation in conditional densities $f(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}} F(\boldsymbol{x})$, we have

$$f_{[1]}(x_1 \mid \mathsf{X}) f(\mathsf{X}_2 = x_2, \ldots, \mathsf{X}_d = x_d) \tag{2.40}$$
$$= f(x_1 \mid \mathsf{X}_2 = x_2, \ldots, \mathsf{X}_d = x_d) f(\mathsf{X}_2 = x_2, \ldots, \mathsf{X}_d = x_d) \tag{2.41}$$
$$= f(\mathsf{X}_1 = x_1, \mathsf{X}_2 = x_2, \ldots, \mathsf{X}_d = x_d). \tag{2.42}$$

It follows that if $\mathsf{X} \sim F$, $[1](\cdot \mid \mathsf{X}) \overset{\mathrm{D}}{=} F$. We argued this for $F_{[1]}$, but by symmetry the result holds for $F_{[k]},\ k = 1, \ldots, d$. Thus F is a stationary distribution of the Markov chain with associated transition kernel $F_{[k]}(\cdot \mid \cdot)$ for each k. Combining these coordinate-wise kernels by choosing a random coordinate at each step gives overall a homogenous mixture kernel. In particular it is an equally-weighted mixture of the d conditional transition kernels corresponding to each of the coordinates of the variable.

$$F(\cdot \mid \boldsymbol{x}) = \frac{1}{d} \sum_{k=1}^{d} F_{[k]}(\cdot \mid \boldsymbol{x}). \tag{2.43}$$

The samples $\{\boldsymbol{x}^{(m)}\}, m = 1, 2, \ldots$ are by construction a realization of a Markov chain whose stationary distribution is given by F. If the Markov chain corresponding to the sampler is additionally irreducible it is ergodic, and sample averages over this chain converge to the averages of samples drawn from the stationary distribution; that is, it is a consistent estimator. Irreducibility is not in general given for a Gibbs sampler. A sufficient condition to ensure irreducibility, which holds in all examples we consider, is that the support of the chain is connected (Roberts and Smith 1994).

2.5 Importance sampling

Importance sampling (IS) (Rubin 1987) modifies the CMC method for computing expectations with respect to some inconvenient F into one that calculates that expectation using reweighted samples from a different distribution G_τ. We present here the weighted IS estimator in the form in which it is employed in rare event problems (Botev and Ridder 2017; Kong 2014). Our exposition follows Rubino

and Tuffin (2009, Chapter 2).

Consider an estimand of the form (2.5), i.e. $\theta = \mathbb{E}[\zeta(X)]$. Suppose we have a family of distributions $\{G_\tau\}_\tau$ parameterized by τ such that F is absolutely continuous with respect to G_τ for all τ. Then $\mathrm{d}F(x)/\mathrm{d}G_\tau(x)$ is defined on all $x \in \mathrm{supp}(Z)$ for $Z \sim G_\tau$. Suppose further, for ease of exposition, that auxiliary law G_τ and target law F both have densities, respectively, g_τ and f with respect to the Lebesgue measure on $\mathbb{R}^d$. Extension to distributions with atoms is immediate. Importance sampling leverages the observation that

$$\mathbb{E}_{X\sim F}[\zeta(X)] = \int \zeta(x) f(x) \mathrm{d}x = \int \zeta(x) \frac{f(x)}{g_\tau(x)} g_\tau(x) \mathrm{d}x = \mathbb{E}_{Z\sim G_\tau}\left[\zeta(Z)\frac{f(Z)}{g_\tau(Z)}\right]. \tag{2.44}$$

With this we modify the Crude Monte Carlo estimator (2.8) using independent simulations $Z_{(i)} \sim G_\tau$ to find

$$\widehat{\mathbb{E}_{\mathrm{IS}}}[\zeta(X)] = \frac{1}{n}\sum_{i=1}^{n} \frac{f(Z_{(i)})}{g_\tau(Z_{(i)})}\zeta(Z_{(i)}). \tag{2.45}$$

Unlike the basic crude Monte Carlo method (2.8), IS does not require us to simulate exactly from F if we can instead simulate exactly from G_τ and calculate the likelihood ratios $\frac{f(x)}{g_\tau(x)}$. The efficiency of this approach depends on finding a "good" approximating G_τ such that the estimator variance is small. A complete recipe for importance sampling fixes the family of auxiliary laws $\{G_\tau\}_\tau$ and specifies a procedure to select an efficient value for the parameter τ.

Design and analysis of the efficiency of these algorithms in the general case is a field of its own (e.g. Kroese, Taimre, and Botev 2011; Rubino and Tuffin 2009; Vehtari et al. 2019). Achieving good performance often requires a hand-designed analysis to tune parameter τ for each choice of F. For many of the problems we consider the benchmark algorithm is such a minutely-tuned IS estimator. As presaged, our innovation is estimators that are competitive with the state of the art IS estimator without requiring such manual tuning.

2.6 Splitting

Splitting methods are Monte Carlo techniques which sequentially decompose a difficult estimation problem for a given process into several easier ones. Intuitively, such methods entail simulating variates attaining successive nested events constructed such that they converge in some sense to the desired target distribution restricted to the event of interest. Methods based on the splitting idea are the central concern of our subsequent analysis. We wish to estimate quantities of the form (2.2) or (2.1). This can be either a rare-event-truncated integral $\theta = \phi(X)|\mathcal{L}$ and/or rare-event truncated probability $\mathbb{P}[\mathcal{L}]$ where $\ell = \mathbb{P}[\mathcal{L}] \ll 1$. The splitting method provides an unbiased estimate $\hat{\ell}$, and a consistent estimate of rare-event conditional estimands $\hat{\theta}$ (Botev and L'Ecuyer 2020; L'Ecuyer, Botev, and Kroese 2018).

2.6.1 Basic splitting

Splitting methods originate in the physics simulation literature (Kahn and Harris 1951). Their modern popularity dates to the introduction in altered form of *RESTART* methods (Villén-Altamirano and Villén-Altamirano 1991; Villén-Altamirano and Villén-Altamirano 1994). Since that landmark they have been analysed and extended by many authors. Efficiency and optimality analysis of various splitting methods can be found in Cérou et al. (2006), Dean and Dupuis (2009), Glasserman et al. (1998a), and Glasserman et al. (1998b). Adaptive versions are considered by Bréhier, Lelièvre, and Rousset (2015), Cérou and Guyader (2007), Cérou and Guyader (2016), and Charles-Edouard et al. (2015). Although the original splitting method was applied strictly to time-indexed Markov stochastic processes, it has been extended to broader settings (Botev and Kroese 2012; Botev et al. 2012). Comprehensive synthesis of much of the research may be found in the thesis by Garvels (2000). Connections between splitting methods and Sequential Monte Carlo (Del Moral, Doucet, and Jasra 2006; Doucet, Freitas, and Gordon 2001) have been made by Cérou et al. (2006), Del Moral, Doucet, and Jasra (2006), Del Moral and Lezaud (2006), Johansen, Moral, and Doucet (2006), and L'Ecuyer et al. (2009). This has led to central limit theorems for splitting

methods (Cérou et al. 2006; Del Moral and Lezaud 2006; Johansen, Moral, and Doucet 2006; Le Gland and Oudjane 2006) which we use below. Splitting methods are now established enough to feature as a staple of modern simulation method textbooks, e.g. (Rubinstein and Kroese 2016, p. V.5; Kroese, Taimre, and Botev 2011, Ch 9; Asmussen and Glynn 2007, Ch 10; Rubino and Tuffin 2009, Ch 3).

The splitting method aims to sample some rare-event-truncated set by successively simulating realizations of a some sequentially-dependent random objects through a succession of increasingly rare target events. The motivating intuition is that, although a CMC estimator might have poor relative error in calculating our target event, possibly we can use instead a series of conditionally less-rare intermediate events. The hope is that this decomposes the hard problem into a series of easier problems such that the overall procedure is easier. A general version of the method is elaborated in Rubinstein and Kroese (2016, Ch 9): we look for a sequence of intermediate random variables, $X^{(m)}$, and a corresponding sequence of events $X^{(m)} \in \mathcal{L}^{(m)}$ for $m = 1, 2, \ldots, M$, such that these events are nested:

$$\mathcal{L}^{(1)} \supseteq \mathcal{L}^{(2)} \supseteq \cdots \supseteq \mathcal{L}^{(M)}. \tag{2.46}$$

We consider the *conditional survival probabilities*

$$p^{(m)} = \begin{cases} \mathbb{P}[\mathcal{L}^{(1)}] & \text{if } m = 1 \\ \mathbb{P}[\mathcal{L}^{(k)} | \mathcal{L}^{(k-1)}], & m = 2, \ldots, M. \end{cases} \tag{2.47}$$

We note that under the nesting condition (2.46)

$$\mathbb{P}[\mathcal{L}] = \ell = \prod_{m=1}^{M} p^{(m)}. \tag{2.48}$$

If each of these conditional survival probability problems is not a rare event problem so that $p^{(m)} \gg \ell$, for $m = 1, 2, \ldots, M$, then we hope that estimating each of them is in some sense tractable.

The splitting method constructs an empirical estimator $\hat{p}^{(1)}, \hat{p}^{(2)}, \ldots, \hat{p}^{(M)}$ of

the conditional-survival-probabilities by simulation. The estimator of ℓ, is

$$\hat{\ell} = \prod_{m=1}^{M} \hat{p}^{(m)}. \tag{2.49}$$

We examine the properties of this estimator momentarily, after detailing the basic algorithm.

The intermediate random variables $\{\boldsymbol{X}^{(m)}; m = 1, \ldots, M\}$ are usually presumed to take values in some common state space, but this is not required. We require that we can simulate from the successive conditional laws, $F^{(1)}(\cdot)$ and $F^{(m)}(\cdot \mid \boldsymbol{X}^{(k)}), k < m]$ for $m = 2, \ldots, M$. We call these conditional distributions *evolution* distributions.

In the splitting method we maintain at each step m an ordered array $\mathcal{X}^{(m)}$ of particles generated by the same stochastic procedure, which we call a *population*. When we wish to discuss, specifically, the samples realized from these random variables, we write $\boldsymbol{\xi}^{(m)}$, where each such $\boldsymbol{\xi}^{(m)}$ is sampled from $\boldsymbol{\xi}^{(m)} \sim F^{(m)}(\cdot \mid \boldsymbol{X}^{(m-1)} = \boldsymbol{\xi}^{(m-1)})$. These objects are segments of the trajectory of some random process in the state space $\mathbb{R}^d$. We follow the Sequential Monte Carlo literature in referring to such realizations as *particles*. *Chains* and *realizations* are also common terms. Individual particles in a population are referenced by a subscript (i), for example $\boldsymbol{\xi}_{(i)}^{(m)} \in \mathcal{X}^{(m)}$. We use $\mathcal{I}(\mathcal{X})$ to denote an index set over all such particles for a given $\mathcal{X}$.

The algorithm proceeds iteratively. Starting with the initial population $\tilde{\mathcal{X}}^{(0)}$ comprising $\tilde{n}^{(0)}$ copies of $\boldsymbol{X}^{(1)}$, we discard (or "prune") any that are not contained in $\boldsymbol{\xi} \notin \mathcal{L}^{(1)}$ and record the population of remaining particles as $\mathcal{X}^{(1)}$. We record the size of the surviving population as $n^{(1)} = \left|\mathcal{X}^{(1)}\right|$. The estimate for the first conditional survival probability is $\hat{p}^{(1)} = n^{(1)}/\tilde{n}^{(0)}$. We draw a resampling vector $\boldsymbol{r} = (r_1, r_2, \ldots) \sim R(\cdot; n^{(m)})$ (to be discussed momentarily) and use it to clone the surviving particles in $\mathcal{X}^{(1)}$ by resampling. The new population is $\tilde{\mathcal{X}}^{(1)} = \{\boldsymbol{\xi}_{r_1}, \boldsymbol{\xi}_{r_2}, \ldots\}$. We record the population count $\tilde{n}^{(1)} = \left|\tilde{\mathcal{X}}^{(1)}\right|$. We conceptualize this resampling as *splitting* the sample paths.

Now, for each successive step $m = 2, 3, \ldots, M$ we repeat a similar procedure. Each particle in $\boldsymbol{\xi}^{(m)} \in \tilde{\mathcal{X}}^{(m)}$ is evolved forward independently, simulating from the conditional evolution distributions, $\boldsymbol{\xi}^{(m)} \sim F^{(m)}(\cdot \mid \boldsymbol{X}^{(m-1)}{=}\boldsymbol{\xi}^{(m-1)})$. We prune

any particles that are not contained in the desired intermediate target set $\boldsymbol{\xi} \notin \mathcal{L}^{(m)}$ and record the remaining population of particles as $\mathcal{X}^{(m)}$. The size of the surviving population we record as $n^{(m)} = \left|\mathcal{X}^{(m)}\right|$. We calculate $\hat{p}^{(m)} \stackrel{\text{def}}{=} n^{(m)}/\tilde{n}^{(m-1)}$ and $\hat{\ell}^{(m)} = \prod_{k=1}^{m} \hat{p}^{(m)}$ (2.49) as an estimate of $\mathbb{P}[\mathcal{L}^{(m)}]$. This whole procedure is elaborated in Algorithm 2.1.

The resampling distributions in splitting, unlike the evolution distributions, do not depend upon the states $\boldsymbol{\xi}$ in the following sense. The splitting method at each step chooses a resampling of the population, which is to say, the resampling distributions $R(\cdot; n^{(m)})$ over resampling vectors describe random resamplings whose realizations are rearrangements of lists. A realization of a resampling gives us the (possibly duplicated) indices of the population to be selected for the next time step, $R : \Omega \to \{1, 2, \ldots, n^{(m)}\}^{|\tilde{N}^{(m+1)}|}$. Here $\tilde{N}^{(m+1)}$ may itself be a random variable. We allow R to depend upon the particles only through the cardinality of the population $N^{(m)}$ and the step index m, but not the *values* of the particles in the population list.

Popular choices for R include *fixed factor* splitting, and *fixed effort* splitting. In *fixed factor* splitting, each particle is duplicated a fixed (deterministic) number K_{FF} of times, e.g. for $K_{\text{FF}} = 3$ we would have a deterministic resampling $R^{(m)} \sim R_{\text{FF}} \Rightarrow \mathbb{P}[\boldsymbol{r} = [1, 1, 1, 2, 2, 2 \ldots, n^{(m-1)}, n^{(m)}, n^{(m)}, n^{(m)} = 1]$. In *fixed effort* splitting, a fixed population count $\tilde{n} \stackrel{\text{def}}{=} \tilde{n}^{(0)} = \tilde{n}^{(1)} = \cdots = \tilde{n}^{(M-1)}$ is maintained by resampling with replacement uniformly at random. In particular this formulation of fixed effort is called *fixed effort with random assignment using multinomial resampling*, or *bootstrap resampling* in the literature. In our case, this means for $K_{\text{FE}} = 3$ we would have $R^{(m)} \sim R_{\text{FE}} = \text{Unif}(\{1, \ldots, N^{(m)}\})^{\times \tilde{n}}$. Here $\text{Unif}(\mathcal{X})$ is the discrete uniform distribution over the finite set $\mathcal{X}$. In effect, this means that we renew the population by sampling from it uniformly at random with replacement, i.e., bootstrap (Efron 1979) resampling. For our purposes, we assume fixed effort splitting with bootstrap resampling throughout. In this fixed effort case, a natural effort parameter is $\eta = \tilde{n}M$, which counts the total number of random realizations.

A virtue of splitting estimators is that, under mild conditions, Algorithm 2.1 $\hat{\ell} = \prod_{m=1}^{M} \hat{p}^{(m)}$ yields an unbiased estimator $\mathbb{E}[\hat{\ell}^{(m)}] = \mathbb{P}[\mathcal{L}^{(m)}]$ of $\ell = \mathbb{P}[\mathcal{L}]$. In fact $\mathbb{E}[\ell^{(m)}] = \mathbb{P}[\mathcal{L}^{(m)}]$ for each m (e.g. Asmussen and Glynn 2007, ch 9). For rare-event *conditional* estimands, the situation is more complicated and we have fewer

Algorithm 2.1 Generalized splitting

Require: Initial population count $\tilde{n}^{(0)}$.
Require: Target nested events $\mathcal{L}^{(1)}, \ldots, \mathcal{L}^{(M)}$, such that $\boldsymbol{X}^{(m-1)} \notin \mathcal{L}^{(m-1)} \Rightarrow \boldsymbol{X}^{(m)} \notin \mathcal{L}^{(m)}$

Simulate: $\tilde{\mathcal{X}}^{(1)} \leftarrow \{\boldsymbol{\xi}_{(i)}^{(m)} \sim F^{(1)} : i = 1, \ldots, \tilde{n}^{(0)}\}$
Prune: $\mathcal{X}^{(1)} \leftarrow \tilde{\mathcal{X}}^{(1)} \cap \mathcal{L}^{(1)}$
Record pruned population count $n^{(1)} \leftarrow |\mathcal{X}^1|$
$\hat{p}^{(1)} \leftarrow n^{(1)}/\tilde{n}^{(0)}$
$\hat{\ell}^{(1)} \leftarrow \hat{p}^{(1)}$
for all $m = 2, 3, \ldots, M$ **do**
    Choose resample index vector $[r_1, \ldots, r_{\tilde{n}^{(m)}}] = \boldsymbol{r} \sim R(\cdot; n^{(m)})$
    Split: $\tilde{\mathcal{X}}^{(m)} \leftarrow \{r_i\text{th particle of } \mathcal{X}^{(m)} \text{ for each } r_i \in \boldsymbol{r}\}$.
    Record population count $\tilde{n}^{(m)} \leftarrow |\tilde{\mathcal{X}}^{(m-1)}|$
    Simulate forward: $\mathcal{X}^{(m)} \leftarrow \{\boldsymbol{\xi}_{(i)}{}^{(m)} \sim F^{(m)}(\cdot \mid \boldsymbol{X}^{(m-1)} = \boldsymbol{\xi}_{(i)}{}^{(m-1)})$ for $i \in \mathcal{I}(\tilde{\mathcal{X}}^{((m-1))})\}$)
    Prune: $\mathcal{X}^{(m)} \leftarrow \mathcal{X}^{(m)} \cap \mathcal{L}^{(m)}$
    Record pruned population count $n^{(m)} \leftarrow |\mathcal{X}^{(m)}|$
    if $\mathcal{X}^{(m)} = \emptyset$ **then**
        return 0
    $\hat{p}^{(m)} \leftarrow n^{(m)}/\tilde{n}^{((m-1))}$
    $\hat{\ell}^{(m)} \leftarrow \hat{p}^{(m)}\hat{\ell}^{(m-1)}$
return $\hat{\ell}^{(M)}$, an estimate of $\ell = \mathbb{P}[\mathcal{L}^{(M)}]$

concrete results. We examine the estimator distribution with such estimands in Subsection 2.6.4.

The category of Generalized Splitting includes a vast number of possible estimators, and we specialize in practical problems. Exploring one useful implementable family of algorithms and problem structures comprises the bulk of the work in Part I.

2.6.2 Dynamic splitting

Before we describe these, we introduce a concrete family of splitting methods, known as *dynamic splitting*. Dynamic splitting is named for its applicability to dynamic problems, by which we mean, problems whose target events are characterized by the states of a time-indexed process $\{X(t)\}_t$. We contrast these dynamic problems with *static problems* wherein the process of interest has no (*a priori*) time index. Dynamic splitting methods are historically prior to the Generalized Splitting method (Kahn and Harris 1951; Villén-Altamirano and Villén-Altamirano 1991). Within the realm of dynamic splitting methods there are many variations, depending on the process of interest, the nature of the target event, and details of the splitting procedure. Overviews are available in Rubinstein and Kroese (2016, p. V.5), Kroese, Taimre, and Botev (2011, Ch 9), Asmussen and Glynn (2007, Ch 10), and Rubino and Tuffin (2009, Ch 3).

We give here the simplest formulation sufficient to our current purposes. The process of interest $\{X(t)\}_{0\le t\le t_{\max}} \sim \mu$ is Markov, i.e. $X(u) \perp\!\!\!\perp X(s) \mid X(t)$ for any $s < t < u$. We assume that we may sample path segments from the distribution of $\{X(t)\}_{t\in(s,u]}$ over intervals $(s, u]$ conditional on the value of the path at the start of the interval.[5] We write $\mu(\cdot \mid X(s))$ for the associated conditional distribution over paths $\{X(t)\}_{s<t} \mid X(s)$. We refer to a realized path sampled from the distribution of the process of interest over some interval $I \subset \mathbb{R}$ as $\{\boldsymbol{\xi}(t)\}_{t\in I}$. Realized paths comprise the particles in the population at each of the intermediate splitting steps. The intermediate target events are defined with reference to a series of *splitting*

[5]Properly speaking, discussing μ as a "distribution over paths" requires us to specify an underlying probability space, stipulate our measurable events and so on. We are for the moment assuming the existence of an appropriate underlying measure space. We shortly dispense with the need for distributions over entire paths, and return to fixed sets of distributions, however.

times, $t_1, \ldots, t_M$. In terms of Generalized Splitting, the particles in Algorithm 2.1 are vectors of such path segments over intervals. Since this is a splitting method, intermediate target events must be nested (2.46). Typically, the events are such that the paths of the process are not permitted to escape from some target sets in state space before a given time; we would choose, say, $A^{(m)} \subset \mathbb{R}^d, m = 1, 2, \ldots, M$ with $A^{(M)} \subseteq A^{(M-1)} \subseteq \cdots \subseteq A^{(1)}$, and define

$$\mathcal{L}^{(m)} \stackrel{\text{def}}{=} \{\boldsymbol{X}(t) \in A^{(m)}\}_{t \in [0, t_m]}. \tag{2.50}$$

Such events are by construction nested. The specialization of the Generalized Splitting algorithm induced by such a structure is expanded in Algorithm 2.2.

Algorithm 2.2 Simple dynamic splitting

Require: Initial population count $\tilde{n}^{(0)}$.
Require: Splitting times $t_1 < \cdots < t_M = t_{\max}$.
Require: Nested target events $\mathcal{L}^{(m)}, \ldots, \mathcal{L}^{(M)}$.
Simulate particles' states $\tilde{\mathcal{X}}^{(1)} \leftarrow \{\{\boldsymbol{\xi}(t)_{(i)}\}_{t \in (0, t_1]} \sim F(\cdot; t_1) : i = 1, \ldots \tilde{n}^{(0)}\}$
Prune: $\mathcal{X}^{(1)} \leftarrow \left\{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\mathcal{X}^{(m)}) \text{ if } \boldsymbol{\xi}_{(i)} \in \mathcal{L}^{(1)} = \{S(\boldsymbol{\xi}_{(i)}(t)) \leq \kappa,\ \forall t \leq t_1\}\right\}$
Record pruned population count $n^{(1)} \leftarrow |\mathcal{X}^1|$
$\hat{p}^{(1)} \leftarrow n^{(1)} / \tilde{n}^{(0)}$
$\hat{\ell}^{(1)} \leftarrow \hat{p}^{(1)}$
for all $m = 2, 3, \ldots, M$ **do**
    Choose resample index vector $[r_1, \ldots, r_{\tilde{n}^{(m)}}] = \boldsymbol{r} \sim R(\cdot; n^{(m)})$
    Split states $\tilde{\mathcal{X}}^{(m)} \leftarrow \{r_i\text{th particle of } \mathcal{X}^{(m)} \text{ for each } r_i \in \boldsymbol{r}\}$.
    Record initial population count $\tilde{n}^{(m)} \leftarrow |\tilde{\mathcal{X}}^{((m-1))}|$
    Simulate forward: $\mathcal{X}^{(m)} \leftarrow \{\{\boldsymbol{\xi}_{(i)}(t)\}_{t \in (t_{m-1}, t_m]} \sim F(\cdot \mid \boldsymbol{\xi}_{(i)}(t_{m-1}); t_m) : \boldsymbol{\xi}_{(i)} \in \tilde{\mathcal{X}}^{((m-1))}\}$
    Prune: $\mathcal{X}^{(m)} \leftarrow \left\{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\mathcal{X}^{(m)}) \text{ if } \boldsymbol{\xi}_{(i)} \in \mathcal{L}^{(m)}\right\}$
    Record pruned population count $n^{(m)} \leftarrow |\mathcal{X}^{(m)}|$
    if $\mathcal{X}^{(m)} = \emptyset$ **then**
        return $0, \emptyset$
    $\hat{p}^{(m)} \leftarrow n^{(m)} / \tilde{n}^{((m-1))}$
    $\hat{\ell}^{(m)} \leftarrow \hat{p}^{(m)} \hat{\ell}^{(m-1)}$
return $\hat{\ell}$ an estimate of $\mathsf{X} \in \mathcal{L}$
return $\mathcal{X}^{(M)}$, approximate samples from the distribution of $\mathsf{X}(t_{\max}) \mid \mathcal{L}$

There is a particularly useful class of target events which we employ henceforth: target events defined as the set of paths of the process of interest such that some function $S : \mathbb{R}^d \to \mathbb{R}$ of a Markov process's state remains below a certain level.[6] We call S an *importance function*. With respect to this function, the target event is given

$$\mathcal{L} = \{S(\boldsymbol{X}(t)) \leq \kappa,\ \forall t < t_{\max}\}. \tag{2.51}$$

A sequence of intermediate target events for such a problem is given by choosing an importance function S, splitting times $0 < t_1 < \cdots < t_M = t_{\max}$ and splitting levels $\kappa^{(1)}, \kappa^{(2)}, \ldots, \kappa^{(M)} = \kappa$. Then, choice of intermediate target events is given

$$\mathcal{L}^{(m)} = \{S(\boldsymbol{X}(t)) \leq \kappa^{(m)},\ \forall t \leq t_m\}. \tag{2.52}$$

This corresponds to a state-based intermediate target level (2.50) with the target sets defined as $A^{(m)} \stackrel{\text{def}}{=} \{x : S(x) \leq \kappa^{(m)}\}$. It remains to the user to ensure the intermediate target events are in fact nested as per (2.46).

2.6.3 Quasi-monotonicity

We introduce a property which is useful in the next example and in the sequel.

Definition 2.5 (Quasi-monotonicity)**.** A function $S : \mathbb{R}^d \to \mathbb{R}$ is *quasi-monotone increasing* in its vector argument for any element index $1 \leq k \leq d$ and any $\delta \geq 0$,

$$S\left(\begin{bmatrix} X_1 \\ \vdots \\ X_k \\ \vdots \\ X_d \end{bmatrix}\right) \geq S\left(\begin{bmatrix} X_1 \\ \vdots \\ X_k + \delta \\ \vdots \\ X_d \end{bmatrix}\right). \tag{2.53}$$

If the last inequality is reversed, we say it is *quasi-monotone decreasing*. If it is either of those, we say simply that it is *quasi-monotone*.

[6] Target events are often more general and can include complicated functions of the path, e.g. attaining a particular target state set $\mathcal{B} \subset \mathbb{R}^d$ before hitting some sink set $\mathcal{C}$ by the terminal time. We do not require such machinery here.

Hereafter we take *decreasing* (resp. *increasing*) to mean *non-increasing* (resp. *non-decreasing*).

Example 2.8 (Quasi-monotone process splitting)**.** A simple concrete example of a dynamic splitting problem is given by the following structure. Consider a Markov process $\{\boldsymbol{X}(t)\}_{t\in[0,t_{\max}]}$ taking values in $\mathbb{R}^d$ whose paths which are a.s. coordinate-wise increasing. For concreteness we can assume this to be a subordinator (Definition B.2) such as a Gamma process (Section B.2). We assume $\mathbb{P}[\boldsymbol{X}(0) = \mathbf{0}] = 1$. This target event is defined by the importance function exceedance, (2.52), as $\mathcal{L} = \{S(\boldsymbol{X}(t)) \leq \kappa,\ \forall t \leq t_{\max}\}$. We set the importance function S to be quasi-monotone increasing. For this problem we use the natural intermediate target sets $\mathcal{L}^{(m)} = \{S(\boldsymbol{X}(t)) \leq \kappa^{(m)},\ \forall t \leq t_m\}$. A graphical depiction of a realization of a splitting procedure in such a process is given in Figure 2.1.

Useful properties arise from the structure of the quasi-monotone subordinator example. Note that a particle cannot "return" to the target set once it has left, because the importance levels are increasing in each coordinate of the process, and each coordinate is increasing in time. Then, for a target event $\mathcal{L}$ of the dynamical splitting form (2.51),

$$\begin{bmatrix} X_1 \\ \vdots \\ X_k \\ \vdots \\ X_d \end{bmatrix} \notin \mathcal{L} \Rightarrow \begin{bmatrix} X_1 \\ \vdots \\ X_k + \delta \\ \vdots \\ X_d \end{bmatrix} \notin \mathcal{L}. \tag{2.54}$$

It follows that the following events are nested

$$\{S(\boldsymbol{X}(s)) \leq \kappa'\} \subseteq \{S(\boldsymbol{X}(t)) \leq \kappa'\} \tag{2.55}$$

for all κ' and $s \leq t$. Similarly, the following events are equal

$$\{S(\boldsymbol{X}(s)) \leq \kappa',\ \forall s \leq t\} = \{S(\boldsymbol{X}(s)) \leq \kappa'\} \tag{2.56}$$

for all κ' and t. In particular, setting $\kappa' = \kappa$ and $t = t_{\max}$ we see that if we wish to

know whether a particle has remained in the target set over the interval $(t_{m-1}, t_m]$ it suffices to inspect the value of that path at the last instant t_m. By the same logic, the intermediate target events $\mathcal{L}^{(m)}$ are $\boldsymbol{X}(t_m)$-measurable.

We allowed the possibility above that the intermediate levels $\kappa^{(m)}$ were different. In fact, in such a problem we should set the levels to be identical across all intermediate target events, $\kappa^{(1)} = \kappa^{(2)} = \cdots = \kappa^{(M)} = \kappa$. This follows from the fact that if $\kappa^{(2)} > \kappa^{(1)}$ then we no longer have in general that $\mathcal{L}^{(1)} \supseteq \mathcal{L}^{(2)}$ so this possibility is excluded. If, on the other hand, we have $\kappa^{(2)} < \kappa^{(1)}$ then the sets are still nested $\mathcal{L}^{(1)} \supseteq \mathcal{L}^{(2)}$, but we are admitting particles $S(\boldsymbol{X}(t_1)) > \kappa^{(2)}$ which we know will be killed at some later step, and so we gain nothing by expending effort in simulating their trajectories. It follows that we choose $\kappa^{(2)} = \kappa^{(1)}$. We may by induction argue that all $\kappa^{(m)}$ must be equal. Such target events are nested, since

$$\mathcal{L}^{(m)} = \{S(\boldsymbol{X}(t_m) \leq \kappa\} \subseteq \{S(\boldsymbol{X}(t_{m-1}) \leq \kappa\} = \mathcal{L}^{(m-1)}. \tag{2.57}$$

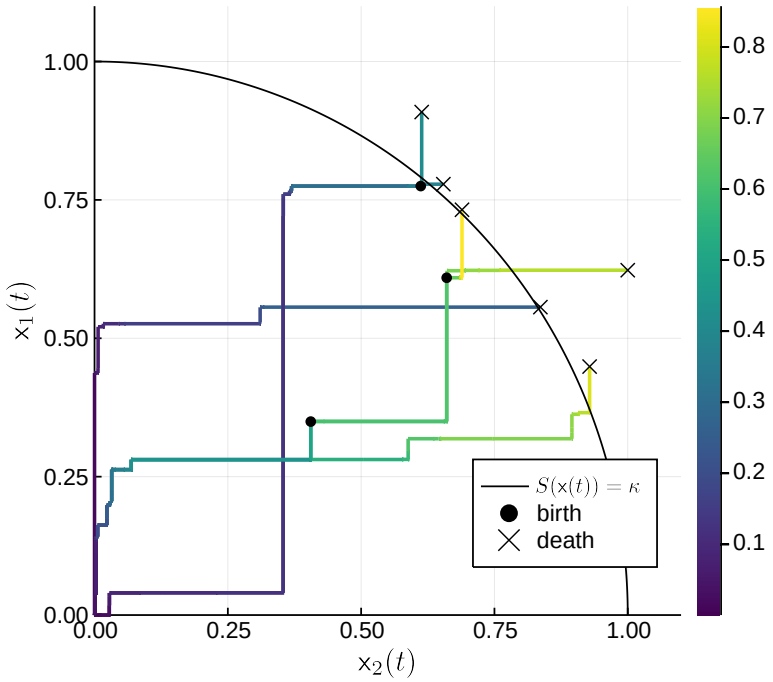
We revisit this example in depth in Chapter 3 where we make extensive use of these properties.

2.6.4 Distribution of splitting estimates

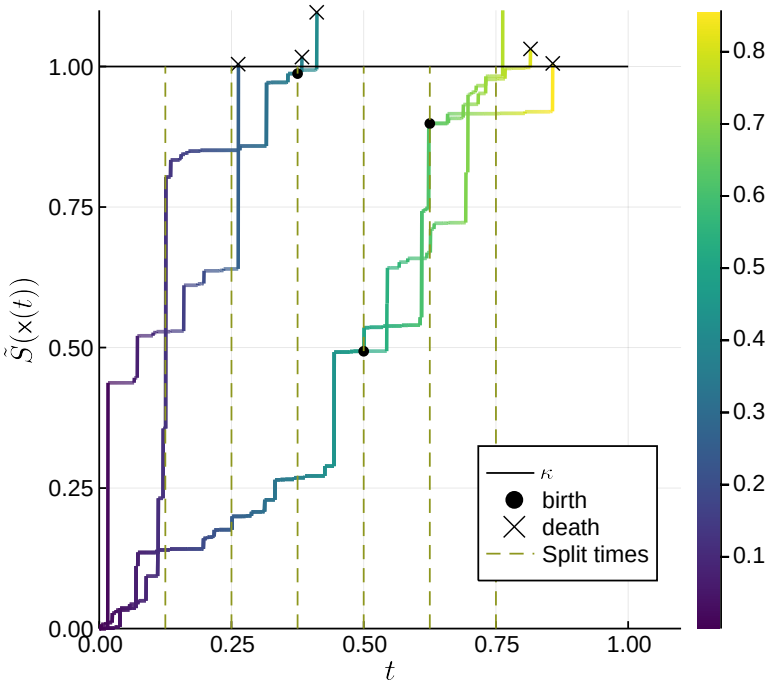
In the dynamic splitting method we consider the distributions $\mu^{(m)}$ of states of the stochastic process $\boldsymbol{X}(t_m)$ associated with each given target event $\mathcal{L}^{(m)}$ conditional upon that event, i.e.

$$(\boldsymbol{X}(t_m) \mid \mathcal{L}^{(m)}) \sim \mu^{(m)}. \tag{2.58}$$

We call these *entrance distributions*. Of particular interest is the final entrance distribution, $\mu^{(M)}$ from whose distribution are drawn the samples that form our ultimate estimator.



(a) Paths of process



(b) Importance function

Figure 2.1: Two equivalent depictions of simple dynamic splitting of a bivariate coordinate-wise increasing process (Example 2.8). Here both components of X are independent univariate gamma subordinators, Section B.2 $X \sim \text{GammaProc}(6, \frac{1}{4})^{\times 2}$, threshold κ=1 and S:$\boldsymbol{x} \mapsto \|\boldsymbol{x}\|_2$. Splitting levels are all equal, $\kappa^{(1)} = \kappa^{(2)} = \cdots = \kappa^{(M)} = \kappa = 1$. Colour of paths denotes time.

Distribution of rare-event-probability estimates

The following derived estimator is unbiased for $m = 1, \ldots, M$ (L'Ecuyer et al. 2009, p. 44), i.e. for rare-event-truncated estimands (2.1):

$$\hat{\mathbb{E}}[\phi(\boldsymbol{X}(t_m))\mathbb{I}\{\mathcal{L}^{(m)}\}] \stackrel{\text{def}}{=} \hat{\ell}\frac{1}{n^{(m)}} \sum_{i \in \mathcal{I}(\mathcal{X}^{(m)})} \phi(\boldsymbol{\xi}_{(i)}(t_m)). \tag{2.59}$$

In particular, as we presaged in the case of Generalized Splitting, estimates of $\mathbb{P}[\mathcal{L}]$ are unbiased.

Analysis of the higher moments of the estimator distribution is more involved. For fixed factor splitting, the estimator variance is studied in, for example, Botev, L'Ecuyer, and Tuffin (2012), Dean and Dupuis (2009), Glasserman et al. (1998a), and Glasserman et al. (1999). In the approach used here, fixed *effort* splitting, we leverage the central limit theorems of Cérou et al. (2006). These use the interacting particle systems formalism of Del Moral (2004) to derive large-effort asymptotic results. In estimating ℓ specifically the following central limit theorem is useful (e.g. Garvels and Kroese 1998; Garvels, Ommeren, and Kroese 2002; L'Ecuyer, Demers, and Tuffin 2006):

Proposition 2.1 (Central limit theorem for rare-event truncated expectation by fixed-effort splitting (Cérou et al. 2006; Chan and Lai 2013; L'Ecuyer et al. 2009))**.** We define

$$h_{\mathcal{L}}(x) = \mathbb{P}\left[\mathcal{L} \mid \boldsymbol{X}(t) = x\right] \tag{2.60}$$

and

$$v^{(m)} = \frac{\operatorname{Var}\left[h_{\mathcal{L}}(\boldsymbol{X}(t_m)) \mid \mathcal{L}^{(m)}\right]}{\mathbb{E}^2\left[h_{\mathcal{L}}\left(\boldsymbol{X}\left(t_m\right)\right) \mid \mathcal{L}^{(m)}\right]} \tag{2.61}$$

$$= \frac{\int_E h_{\mathcal{L}}^2(x)\mathrm{d}\mu^{(m)}(x)}{\left(\int_E h_{\mathcal{L}}(x)\mathrm{d}\mu^{(m)}(x)\right)^2} - 1. \tag{2.62}$$

Then, for each $1 \leq m \leq M$, in distribution as $\tilde{n} \to \infty$

$$\sqrt{\tilde{n}}\left(\frac{\hat{p}^{(1)} \cdots \hat{p}^{(m)}}{p^{(1)} \cdots p^{(m)}} - 1\right) \xrightarrow{\mathrm{D}} \mathcal{N}\left(0, \sqrt{V^{(m)}}\right) \tag{2.63}$$

where

$$V^{(m)} = \sum_{k=1}^{m} \left(\frac{1}{p^{(k)}} - 1 \right) + \sum_{k=1}^{m-1} \frac{v^{(k)}}{p^{(k)}}. \tag{2.64}$$

In practice, it is difficult to evaluate the integral (2.62) to find the variance $V^{(m)}$, since it is not clear how to handle these $v^{(m)}$ terms in the general case. The standard argument (e.g. Botev, L'Ecuyer, and Tuffin 2012; Bréhier, Lelièvre, and Rousset 2015; Garvels 2000; Glasserman et al. 1999; Guyader, Hengartner, and Matzner-Løber 2011; L'Ecuyer, Demers, and Tuffin 2006; Lagnoux 2006) instead analyses an "idealized" problem in which we can ignore the contribution of the $v^{(m)}$ terms. We observe that the asymptotic variance term $V^{(m)}$ (2.64) decomposes into two parts, one involving only the conditional survival probabilities $p^{(m)}$, and one term which involves $v^{(m)}$ and thus the distributions of $\{\mathsf{X}(t_m)\}_t$ via the $\mu^{(m)}$ entrance distributions in (2.61). If we can work instead with a problem wherein

$$v^{(1)} = \cdots = v^{(M)} = 0 \tag{2.65}$$

then we can eliminate the troublesome terms. In such an idealized problem we suppose that, for any given m,

$$\operatorname{Var}\left[h_{\mathcal{L}}(\mathsf{X}(t_m)) \mid \mathcal{L}^{(m)} \right] = 0 \tag{2.66}$$

and thus $h_{\mathcal{L}}$ is constant on $\operatorname{supp}(\mu^{(m)})$. In effect, this requirement implies that the survival probability $p^{(m)} = \mathbb{P}[\mathcal{L} \mid \mathsf{X}(t_m) \in \mathcal{L}^{(m)}]$ of any given path of the process depends on the state $\mathsf{X}(t_m)$ only through t_m. In this case, the conditional survival probabilities are given by $p^{(m)} \stackrel{\text{def}}{=} p(t_m)$ for some unknown function $p : [0, t_{\max}] \to [0, 1]$. At each step m the surviving population $n^{(m)}$ may then be regarded as an independent draw of a random variable $n^{(m)} \sim \mathsf{N}^{(m)}$ distributed as

$$\mathsf{N}^{(m)} \sim \text{Binomial}(\tilde{n}, p^{(m)}). \tag{2.67}$$

We call this idealized model the *state-independent* model. For such state-independent models, we can find the form for the large-effort asymptotic variance

using (2.64) as

$$V^{(m)} = \sum_{k=1}^{m} \left(\frac{1}{p^{(k)}} - 1 \right). \tag{2.68}$$

This is not immediately applicable as we have not chosen $p^{(k)}$. We can also find and analyse the optimal splitting levels in the state independent model. This amounts to minimising the expected squared error loss function

$$L(M, p^{(1)}, \ldots, p^{(M)}) \stackrel{\text{def}}{=} \frac{\eta}{\ell^2} \operatorname{Var}[\hat{\ell}] \tag{2.69}$$

$$= \frac{M\tilde{n}}{\ell^2} \operatorname{Var}[\hat{p}^{(1)} \ldots \hat{p}^{(M)}] \tag{2.70}$$

subject to

$$\prod_{m=1}^{M} p^{(m)} = \ell \iff \sum_{m=1}^{M} \log p^{(m)} = \log \ell. \tag{2.71}$$

Recalling that $\hat{p}^{(m)} = \mathsf{N}^{(m)}/\tilde{n}, m = 1, \ldots, M$, we find the variance

$$\operatorname{Var}\left(\hat{p}^{(1)} \ldots \hat{p}^{(M)}\right) = \prod_{m=1}^{M} \mathbb{E}[(\hat{p}^{(m)})^2] - \ell^2 \tag{2.72}$$

$$= \prod_{m=1}^{M} \mathbb{E}\left[\left(\frac{\mathsf{N}^{(m)}}{\tilde{n}} \right)^2 \right] - \prod_{m=1}^{M} (p^{(m)})^2 \tag{2.73}$$

$$= \prod_{m=1}^{M} \left(\frac{p^{(m)}(1 - p^{(m)})}{\tilde{n}} + (p^{(m)})^2 \right) - \prod_{m=1}^{M} (p^{(m)})^2. \tag{2.74}$$

We have used here the fact that, as a binomial variate, $\mathbb{E}[(\mathsf{N}^{(m)})^2] = p^{(m)}\tilde{n}(1 - p^{(m)} + \tilde{n}p^{(m)})$.

If we introduce the assumption that fortuitously, for some ideal target survival probability $\check{p}$, $\ell = \check{p}^M$ for $M \in \mathbb{N}$ then we may argue from invariance of (2.74) with respect to exchanges in m, that

$$p^{(1)} = \cdots = p^{(M)} = \check{p}. \tag{2.75}$$

Further, in this case, we have

$$\check{p} = \ell^{1/M}. \tag{2.76}$$

In this case we may simplify (2.69) to

$$\operatorname{Var}[\hat{\ell}] = \left(\check{p}(1-\check{p})/\tilde{n} + \check{p}^2\right)^M - \check{p}^{2M} \tag{2.77}$$

$$= \frac{M\check{p}^{2M-1}(1-\check{p})}{\tilde{n}} + \frac{M(M-1)\check{p}^{2M-2}(1-\check{p})^2}{\tilde{n}^2} + \cdots + \frac{(\check{p}(1-\check{p}))^M}{\tilde{n}^M}. \tag{2.78}$$

If we assume that $\tilde{n}$ grows fast enough that $\tilde{n} \gg M(1-\check{p})/\check{p}$ then all the terms apart from the first are negligible. Discarding them, we find

$$\lim_{\tilde{n}\to\infty} \operatorname{ENRV}(\hat{\ell}) = \lim_{\tilde{n}\to\infty} \frac{M\tilde{n}}{\ell^2} \operatorname{Var}[\hat{\ell}] \tag{2.79}$$

$$= \frac{M^2}{\ell^2}\check{p}^{2M-1}(1-\check{p}) \tag{2.80}$$

$$= M^2\ell^{-1/M}(1-\ell^{1/M}) \tag{2.81}$$

$$= M^2\ell^{-1/M} - M^2\ell \tag{2.82}$$

$$\nearrow M^2\ell^{-1/M} \text{ as } \ell \to 0. \tag{2.83}$$

Differentiating (2.83) we find

$$\frac{\mathrm{d}}{\mathrm{d}M} M^2\ell^{-1/M} = \ell^{-1/M}(2M - \log\ell) \tag{2.84}$$

whose zero at $M = -\frac{1}{2}\log\ell$, turns out to be a minimiser — which is, by assumption, an integer. This gives us our ideal M. Back-substituting into (2.76) we get

$$\check{p} = 1/\mathrm{e}^2 \approx 0.1353. \tag{2.85}$$

The corresponding effort-normalized relative variance is

$$\operatorname{ENRV}(\hat{\ell}) \approx \frac{(\log\ell)^2\mathrm{e}^2}{4}. \tag{2.86}$$

By comparison, the fixed splitting case (Garvels 2000; L'Ecuyer et al. 2009; Lagnoux 2006) yields $\operatorname{ENRV}(\hat{\ell}) \approx 1.5449(\log\ell)^2$. With these optimal parameters, we rewrite the assumption $\tilde{n} \gg M(1-\check{p})/\check{p} = -\frac{1}{2}\log\ell(1-\mathrm{e}^{-2})\mathrm{e}^2$, i.e., $\tilde{n} \gg -\log\ell$.

The resulting variance satisfies

$$\mathrm{Var}[\hat{\ell}] \approx -\frac{\ell^2(\log \ell)^2 \mathrm{e}^2}{2\tilde{n} \log \ell}. \tag{2.87}$$

We are now in a position to examine small-probability efficiency of a splitting estimator constructed using the optimal parameters in the idealized case.

$$\begin{aligned}
\lim_{\ell\to 0} \frac{\log \mathbb{E}\left[\hat{\ell}^2\right]}{\log \ell} &= \lim_{\ell\to 0} \frac{\log\left(\mathrm{Var}[\hat{\ell}] + \ell^2\right)}{\log \ell} && (2.88)\\
&= \lim_{\ell\to 0} \frac{\log\left(\ell^2 \frac{(\log \ell)^2 \mathrm{e}^2}{4\eta} + \ell^2\right)}{\log \ell} && (2.89)\\
&= \lim_{\ell\to 0} \frac{2\log \ell + \log\left(\frac{(\log \ell)^2 \mathrm{e}^2}{-2\tilde{n}\log \ell} + 1\right)}{\log \ell} && (2.90)\\
&= 2 + \lim_{\ell\to 0} \frac{\log\left(-\frac{\mathrm{e}^2 \log \ell}{2\tilde{n}} + 1\right)}{\log \ell} && (2.91)\\
&\approx 2 + \lim_{\ell\to 0} \frac{\log 1}{\log \ell} \text{ if } \tilde{n} \gg -\log \ell && (2.92)\\
&= 2. && (2.93)
\end{aligned}$$

That is, this idealized, asymptotic approximation of our splitting estimator is logarithmically efficient.

Distribution of rare-event-conditional functional estimates

The calculations thus far have concerned the distribution of estimates of probability ℓ. For rare-event-conditional functionals $\theta = \mathbb{E}[\phi(\boldsymbol{X}(t_m)) \mid \mathcal{L}]$ we have fewer results, and the estimator is no longer in general unbiased (Botev and L'Ecuyer 2020; L'Ecuyer, Botev, and Kroese 2018). We have a large-effort central limit theorem that guarantees us consistency and asymptotic normality:

Proposition 2.2 (Central limit theorem for rare-event-conditional expectation estimation by fixed effort splitting)**.** (Cérou et al. 2006). Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be a bounded and continuous function. Let $\boldsymbol{\xi}(t_m)_{(i)}, i \in \mathcal{I}(\mathcal{X}^{(M)})$ denote the values of the paths at time t_m of the surviving particles generated in a splitting estimator

with fixed effort $\tilde{n}$, for $m = 1, \ldots, M$. Then, there is a constant $v(\phi)$, such that

$$\sqrt{\tilde{n}} \left(\frac{1}{n^{(m)}} \sum_{i \in \mathcal{I}(\mathcal{X}^{(m)})} \phi(\xi_{(i)}(t_m)) - \mathbb{E}[\phi(X(t_m)) \mid \mathcal{L}] \right) \xrightarrow{\mathrm{D}} \mathcal{N}(0, \sqrt{v(\phi)}) \tag{2.94}$$

as $\tilde{n} \to \infty$. The result extends to unbounded ϕ under uniform integrability conditions.

The idealization arguments we used to find the optimality of $\check{p}$ and M in the rare-event probability are not directly applicable here, since we essentially assumed there that the survival probabilities of the particles are identical at each step, and thus by construction $\phi(\xi_{(i)}(t_m))$ is constant across the target set and the problem is trivial. Nonetheless, we presume in this case that the optimal $\check{p}$ and M heuristics for the $\hat{\ell}$ estimand are also optimal for the rare-event conditional estimand.

Our arguments about efficiency and consistency of splitting methods have used large-effort asymptotics in various capacities. We might be concerned that in finite-effort settings our estimators are far from the asymptotic distributions. Indeed, investigating numerically we observe that convergence can be different for different estimation problems. An example is shown in Figure 4.8b of estimators converging very slowly to the normal distribution as would be predicted by the Central Limit Theorems. In any case we do not usually have closed form for the variance of the central limit distribution. These asymptotic arguments assure us that the estimators are consistent, but in practice we always check the empirical sampling distribution of the estimator.

By the same token, we might be suspicious of the idealization arguments used to choose $\check{p}$. More general arguments (e.g. Asmussen and Glynn 2007; Botev, L'Ecuyer, and Tuffin 2012; Garvels 2000) propose that reasonable values lie in $\check{p} \in [0.1, 0.5]$. We address this problem with a simulation study in Section 4.3.

Even if we are satisfied with choice of M and $\check{p}$, we do not necessarily know how to construct intermediate target events to $p^{(1)} = \cdots = p^{(M)} = \check{p}$. The selection of the ideal intermediate target events given a choice of optimal conditional survival probabilities $\check{p}$ or more generally $p^{(1)}, \ldots, p^{(M)}$, is referred to in the literature as *level selection*, since it typically reduces to choosing the intermediate target events by choosing levels of importance function, or even the parameters of an importance

function (Garvels, Ommeren, and Kroese 2002). In our case, selecting levels *per se* does not figure. We do still have have the problem of selecting intermediate target events.

2.6.5 Selection of levels

We mention two common alternative approaches to the intermediate target event selection problem. The first approach divides the effort into two runs. The first, called *pilot run*, is not used to construct the pilot estimate, but strictly for intermediate target event selection (e.g. Botev and Kroese 2008; Garvels and Kroese 1998; Glasserman et al. 1999; Villén-Altamirano et al. 1994). The final estimator is constructed from the second, main, run. The alternative approach does a single *adaptive splitting* simulation which calculates optimal parameters online (e.g. Bréhier, Lelièvre, and Rousset 2015; Cérou and Guyader 2007; Cérou and Guyader 2016; Charles-Edouard et al. 2015). Throughout, we pursue the former alternative, although some methods we introduce are in principle also compatible with adaptive splitting. In general, the effort levels required by the pilot run are small enough that the wasted effort is merited by the relative simplicity of analysis.

Improvements to the method of estimating the target sets via a pilot run for our sampling method are our major contribution in this research. This we return to in Chapter 3, where we introduce an algorithm which can cheaply estimate ℓ and M and perform intermediate target event selection to robustly approach ideal performance. In particular, in the quasi-monotone-splitting case we are able to reduce the nebulous problem of intermediate target event selection to an automatically soluble problem of time selection.

Despite the caveats attached to these successive layers of approximation in analysing its performance, the splitting estimator can in practice be a highly efficient estimator in rare event problems. Our own particular splitting variants, for example, do attain near-logarithmic relative efficiency in practice, as we observe in experiments in the sequel.

Chapter 3

Splitting in quasi-monotone problems

In this chapter we introduce the quasi-monotone splitting *method, an efficient splitting Monte Carlo estimator for a family of rare event estimation problems, namely the* quasi-monotone problems. *A key benefit of the proposed method is its wide applicability. It allows for the construction of splitting estimators in a variety of different problems with minimal manual intervention. The method is based on the simple dynamical splitting method for subordinators, introduced in the previous chapter, with a method of mapping certain types of rare event problems onto it. We apply it to rare-event expectations for problems including sums of continuous random variables, partial sums of ordered RVs, ratios of RVs, and weighted sums of Poisson RVs. All these examples are motivated by their practical importance in reliability estimation problems related to wireless communication. We investigate numerically the computational efficiency of the proposed estimator in these problems via a number of simulation studies and find that it compares favourably with existing estimators.*

3.1 Quasi-monotonicity in splitting

The quasi-monotone splitting method generically produces an efficient sampler for a rare event estimation problem by transforming it into a tractable dynamic

splitting problem. The method is generic across a broad class of problems. This chapter is dedicated to identifying the class of problems which can be so treated, showing how the estimator can be constructed, and examining its properties.

For problems amenable to quasi-monotone splitting, the method has a number of desirable features. Unlike, say, Importance Sampling, which requires that the user undertakes problem-specific calculations to choose (approximately) optimal parameters, our method chooses appropriate free parameters automatically. The self-optimized method produces efficient estimators for problems which had none hitherto known, and can be competitive even with specialised Monte Carlo methods.

The quasi-monotone splitting method as developed here targets *static* problems, i.e. ones without an *a priori* time index, which we translate into dynamic splitting problems. The quasi-monotone method is thus an example of *dynamic splitting for static problems*, a specialization of the Generalized Splitting introduced earlier in Section 2.6. The workhorse tool of the method is the dynamic splitting problem introduced in Example 2.8, using a quasi-monotone importance function and dynamic latent process. The quasi-monotone method in fact maps static problems onto members of this family of dynamic splitting problems.

The application of a dynamic splitting method to some static problem $X \in \mathcal{L}$, entails constructing a time-indexed Markov chain whose distribution at a fixed terminal time instant $t_{\max}$ coincides with that of the desired distribution. The random variable X and target event $\mathcal{L}$ is associated with a time-indexed stochastic *process* $\{X(t)\}_{t\in[0,t_{\max}]}$ with terminal marginal distribution

$$(X(t_{\max}) \mid \mathcal{L}') \stackrel{\mathrm{D}}{=} (X \mid \mathcal{L}), \tag{3.1}$$

where $\mathcal{L}'$ is some specific event measurable with respect to the path of the process $\{X(t)\}_{t\in[0,t_{\max}]}$.

3.1.1 Quasi-monotone problems

We recall the setup of quasi-monotone dynamic splitting from Example 2.8. The underlying process in that case was a coordinate-wise increasing[1] process $\{X(t)\}_{t\in[0,t_{\max}]}$ where the ultimate target event was determined by importance function levels, $\mathcal{L}_\kappa \overset{\text{def}}{=} \{S(X(t)) \leq \kappa; \forall t \in [0, t_{\max}]\}$. For that problem, we chose the intermediate target events as $\mathcal{L}^{(m)} = \{S(X(t)) \leq \kappa, \ \forall t \leq t_m\}$. The quasi-monotone process estimator had a variety of desirable features, notably a simple distribution for particle forward simulation at each time step, and a simple representation for the intermediate target sets in terms of time. We give here a set of conditions which enable us to solve non-trivial static problems with the quasi-monotone dynamic splitting method. These we discuss in a "left handed" version. The handedness terminology is explained below.

Definition 3.1 (Left quasi-monotone problems)**.** We say a rare event estimation problem is a *left quasi-monotone problem* if all of the following conditions are satisfied with respect to the rare event problem $X \in \mathcal{L}$.

1. The estimand is of the form (2.1) or (2.2), e.g. $\theta = \mathbb{E}\left[\phi(X)\mathbb{I}\{\mathcal{L}_\kappa\}\right]$ for some $\mathbb{R}^d$-valued random variate X.

2. The target event is defined $\mathcal{L}_\kappa = \{S(X) \leq \kappa\}$ for some importance function $S : \mathbb{R}^d \to \mathbb{R}$.

3. The $\mathcal{L}_\kappa$-conditional distribution $X \mid \mathcal{L}_\kappa$ is equal in distribution to the terminal state of a certain transformed *latent process* $\{G(t)\}_{0\leq t\leq 1}$ taking values on $\mathbb{R}^D$,
$$(X \mid \mathcal{L}_\kappa) \overset{\text{D}}{=} (\rho(G(1)) \mid \mathcal{L}'_\kappa)$$
for some some *recovery function* $\rho : \mathbb{R}^D \to \mathbb{R}^d$, latent process target event $\mathcal{L}'_\kappa$ and $D \geq d$.

4. The target event for the latent process is given
$$\mathcal{L}'_\kappa \overset{\text{def}}{=} \{S_g(G(1)) \leq \alpha(\kappa)\}$$

[1]Recall that we take *decreasing* (resp. *increasing*) to mean *non-increasing* (resp. *non-decreasing*).

for some *latent space importance function* $S_g : \mathbb{R}^D \to \mathbb{R}$ which is quasi-monotone increasing in its argument, and some monotone $\alpha : \mathbb{R} \to \mathbb{R}$.

5. The paths of each component G_k are almost surely coordinate-wise increasing in t,
$$\mathbb{P}[\mathsf{G}_k(u) \leq \mathsf{G}_k(t)] = \mathbb{I}\{u \leq t\} \tag{3.2}$$
with $\mathbb{P}[\mathsf{G}(0) = \mathbf{0}] = 1$

6. S_g is *quasi-monotone increasing* (2.5) in its vector argument.

We introduce some conventions for working with these processes. Although the event $\mathcal{L}'_\kappa$, as the target event of (3.1), is formally different to $\mathcal{L}_\kappa$ of the original problem, hereafter we identify $\mathcal{L}'_\kappa \equiv \mathcal{L}_\kappa$ without ambiguity. When we wish to clarify, we distinguish the latent process $\{\mathsf{G}(t)\}_t$, from the original X by discussing the original random variable as belonging to the *ambient*, as opposed to latent, space. When we write $\mathsf{G}(t)$ we understand it to be referring to the instantaneous value of the process at instant t, $\mathsf{G}(t) \equiv [\mathsf{G}_1(t)\ \mathsf{G}_2(t)\ \ldots\ \mathsf{G}_D(t)]^\top$. We also assume w.l.o.g. that $\mathbb{P}[\mathcal{L}_\kappa] = 0$ when $\kappa \leq 0$ and $\mathbb{P}[\mathcal{L}_\kappa] > 0$ for $\kappa > 0$.

Remark 3.1 (Role of α)**.** We usually set α as the identity mapping, and where we do not state otherwise, this is assumed. This is reasonably general, in that any invertible monotone increasing α may be absorbed into S_g by defining a revised $S'_g \stackrel{\text{def}}{=} S_g \circ \alpha^{-1}$ without changing the quasi-monotone structure of the problem. We see later that the decreasing mapping $\alpha : \kappa \mapsto -\kappa$ allows us to handle such *right*-quasi-monotone problems with $\mathcal{L}_\kappa = \{S(\mathsf{X}) \geq \kappa\}$ in the same setting, as discussed in Definition 3.2. No other forms for α are required here.

Problems satisfying these conditions can be mapped into a variant of the simple dynamic splitting estimator via a transform. The resulting construction aims to simulate from the desired rare-event-conditional random variate $\mathsf{X} \mid \mathcal{L}_\kappa$ by instead simulating realisations of the latent random process $\rho(\mathsf{G}(1)) \mid S_g(\mathsf{G}(1)) < \alpha(\kappa)$ and calculating estimands in the ambient space by taking the transform $\rho(\mathsf{G}(1))$. The splitting method applies to the paths of this latent $\{\mathsf{G}(t)\}_t \mid S_g(\mathsf{G}(t)) < \alpha(\kappa)$. The combined procedure is in Algorithm 3.1.

That this mapping samples from the correct target is immediate — we have

stipulated that in part 3 of the definition. The chief difficulty lies in finding a mapping that satisfies the conditions. We treat this by giving a non exhaustive list of families of problems that do satisfy the conditions. That is to say, we give various sufficient conditions to produce quasi-monotone problems, but not exhaustive necessary conditions. We return to this point momentarily, after expanding upon some of the mechanics.

In a left quasi-monotone problem $\{S_g(\mathsf{G}(t))\}_{0\leq t\leq 1}$ defines a real-valued process with a.s. coordinate-wise increasing paths. By a similar argument to Example 2.8 we see that the target events $\mathcal{L}^{(m)}$ are $\mathsf{G}(t_m)$-measurable. Consider a series of times $0 = t_0 < t_1 < \ldots < t_M = 1$. For any two times $u \geq t$, we have $S_g(\mathsf{G}(u)) \geq \kappa \Rightarrow S_g(\mathsf{G}(t)) \geq \kappa$. It follows that the following events are equal,

$$\mathcal{L}^{(m)} = \{S_g(\mathsf{G}(t)) \leq \kappa,\ \forall t \leq t_m\} = \{S_g(\mathsf{G}(t_m)) \leq \kappa\}. \tag{3.3}$$

In practical terms, this means that the pruning of the particles in the quasi-monotone splitting estimator is easy — we do not need to simulate the entire trajectory $\{\mathsf{G}(t)\}_{t_{m-1}<t\leq t_m} \mid \mathsf{G}(t_{m-1})$ to find if $S_g(\mathsf{G}(t)) \leq \kappa,\ \forall t_{m-1} < t \leq t_m$, since it suffices to simulate the value at a single instant $\mathsf{G}(t_m) \mid \mathsf{G}(t_{m-1})$. For a process which was not quasi-monotone, a practitioner would face considerable manual labour to calculate survival over an interval even in this simple setting. A non-monotonic process can upcross an importance threshold many times and yet be below at the end of an interval.

The structure of the quasimonotone problem may be discussed in a left- or right-handed version. These are nearly interchangeable, although some problems are more naturally expressed in one formulation or the other. We the version presented above *left-handed* since it most naturally handles problems about rare events pertaining to the left tail of a random variable , i.e. where $\mathcal{L}_\kappa = \{S(\mathsf{X}) \leq \kappa\}$.

Definition 3.2 (Right quasi-monotone problems)**.** It is convenient to allow the following alternative formulation. We say a rare event estimation problem is a *right quasi-monotone problem* if it satisfies left quasi-monotone conditions (Definition 3.1), with the following alterations:

2. The target event is defined $\mathcal{L}_\kappa = \{S(\mathsf{X}) \geq \kappa\}$ for some importance function

$S : \mathbb{R}^d \to \mathbb{R}$.

4. The target event for the latent process is given

$$\mathcal{L}'_\kappa \stackrel{\text{def}}{=} \{S_g(\mathsf{G}(1)) \leq \alpha(\kappa)\}$$

for some *latent space importance function* $S_g : \mathbb{R}^D \to \mathbb{R}$ which is quasi-monotone decreasing in its argument, and some monotone $\alpha : \mathbb{R} \to \mathbb{R}$.

6. S_g is *quasi-monotone decreasing* (2.5) in its vector argument.

A right quasi-monotone problem may be transformed into a left-quasi-monotone problem by redefining α and S_g to be $\alpha' : \kappa \mapsto -\alpha(\kappa)$ and $S'_g : \boldsymbol{g} \mapsto -S_g(\boldsymbol{g})$ respectively. In right quasi-monotone problems we assume w.l.o.g. that $\lim_{\kappa\to\infty} \mathbb{P}[\mathcal{L}_\kappa] = 0$ and $\mathbb{P}[\mathcal{L}_0] = 1$. For left quasi-monotone problems we can find a rarity parameter (Definition 2.1) which is an *increasing* function of κ, and for right-quasi-monotone problems, we can find a rarity parameter which is a *decreasing* function of κ. These adjustments do not add expressive power to the left quasi-monotone structure; it is, however, more natural to discuss simulating excess-over-threshold problems such as (2.4) in terms of right-handed quasi-monotone splitting, rather than imagining solving an inverted problem with $\alpha : \kappa \mapsto -\kappa$.

Hereafter we keep notation compact by allowing both left- and right-quasi-monotone problems as convenient. We refer to both classes collectively as quasi-monotone problems. We assume left quasi-monotone structure when describing properties of splitting methods unless otherwise specified, bearing in mind that transforming between them is trivial.

3.1.2 Constructing a mapping

We now return to the question of finding ρ, S_g and so on for particular families of rare-event estimation problems.

A method to find monotone $\mathbb{R} \to \mathbb{R}$ functions with desired target distributions is the well-known *inverse CDF* or *quantile transform* method. We can generate

Algorithm 3.1 Left quasi-monotone splitting

Require: Initial population count $\tilde{n}^{(0)}$.
Require: Splitting times $t_1 < t_2 < \cdots < t_M = 1$.
Require: Target level κ.
Require: Functions S_g, ρ, α and latent process $\{\mathsf{G}\} \sim \{G(\cdot, t)\}$ satisfying quasi-monotone conditions for X and $\mathcal{L}$.
Ensure: $\hat{\ell}$, an estimate of $\mathbb{P}\left(S_g(\mathsf{G}(1)) \leq \alpha\kappa\right) = \mathbb{P}\left(\mathcal{L}\right)$.
Ensure: $\hat{\theta}$, estimate of $\phi(\mathsf{X}) \mid \mathcal{L} = \phi(\rho(\mathsf{G}(1))) \mid S_g(\mathsf{G}(1)) \leq \alpha(\kappa)$.

Simulate latent initial states $\tilde{\mathcal{G}}^{(1)} \leftarrow \{\boldsymbol{\xi}(t_1)_{(i)} \sim G(\cdot; t_1) : i = 1, \ldots \tilde{n}^{(0)}\}$
Prune: $\mathcal{G}^{(1)} \leftarrow \left\{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\mathcal{G}^{(m)}) \text{ if } S_g(\boldsymbol{\xi}_{(i)}(t_1)) \leq \alpha(\kappa), \right\}$.
Record population count $n^{(1)} \leftarrow |\mathcal{G}^1|$.
$\hat{p}^{(1)} \leftarrow n^{(1)}/\tilde{n}^{(0)}$.
$\hat{\ell}^{(1)} \leftarrow \hat{p}^{(1)}$.
for all $m = 2, 3, \ldots, M$ **do**
    Choose resample index vector $[r_1, \ldots, r_{\tilde{n}^{(m)}}] = \boldsymbol{r} \sim R(\cdot; n^{(m)})$.
    Split states $\tilde{\mathcal{G}}^{(m)} \leftarrow \{r_i\text{th particle of } \mathcal{G}^{(m)} \text{ for each } r_i \in \boldsymbol{r}\}$.
    Record population count $\tilde{n}^{(m)} \leftarrow |\tilde{\mathcal{G}}^{(m-1)}|$.
    Simulate latent particles forward: $\mathcal{G}^{(m)} \leftarrow \{\boldsymbol{\xi}_{(i)}(t_m) \sim G(\cdot \mid \boldsymbol{\xi}_{(i)}(t_{m-1}); t_m) : \boldsymbol{\xi}_{(i)} \in \tilde{\mathcal{G}}^{(m-1)}\}$.
    Prune: $\mathcal{G}^{(m)} \leftarrow \{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\mathcal{G}^{(m)}) \text{ if } S_g(\boldsymbol{\xi}_{(i)}(t_m)) \leq \alpha(\kappa)$.
    Record population count $n^{(m)} \leftarrow |\mathcal{G}^{(m)}|$.
    if $\mathcal{G}^{(m)} = \emptyset$ **then**
        return 0.
    $\hat{p}^{(m)} \leftarrow n^{(m)}/\tilde{n}^{(m-1)}$.
    $\hat{\ell}^{(m)} \leftarrow \hat{p}^{(m)}\hat{\ell}^{(m-1)}$.
return $\hat{\ell} = \hat{\ell}^{(M)}$.
return $\hat{\theta} = \frac{1}{\tilde{N}^{(M)}} \sum_{i \in \mathcal{I}(\tilde{\mathcal{G}}^{(M)})} \phi(\rho(\boldsymbol{\xi}_{(i)}))$

any real marginal distribution using an appropriate mapping of a real random variable which is continuous with respect to the real line.

Definition 3.3 (Quantile transform)**.** Suppose our problem requires us to construct a random variate equal in distribution to a real-valued random variable $X \sim F$ from samples of another real-valued random variable $G \sim G$ with $\mathrm{supp}(G) = I \subseteq \mathbb{R}$ where I is a non-trivial interval. If G is absolutely continuous with respect to the Lebesgue measure on $\mathrm{supp}(G)$, we may define the following two quantile transforms:[2]

$$q^{G,F} \stackrel{\text{def}}{=} g \mapsto F^{-1}(G(g)) \qquad \text{the increasing transform.} \tag{3.4}$$

$$q^{G,\bar{F}} \stackrel{\text{def}}{=} g \mapsto \bar{F}^{-1}(G(g)) \qquad \text{the decreasing transform.} \tag{3.5}$$

The naming reflects the fact that the mapping (3.4) is monotone increasing and (3.5) is monotone decreasing.

Noting that

$$G(G(1)) \sim \mathrm{Unif}([0,1]) \tag{3.6}$$

$$1 - G(G(1)) \sim \mathrm{Unif}([0,1]). \tag{3.7}$$

and that $U \sim \mathrm{Unif}([0,1]) \Rightarrow F^{-1}(U) \sim F$, we observe that

$$q^{G,F}(G) \sim F \tag{3.8}$$

$$q^{\bar{G},F}(G) \sim F. \tag{3.9}$$

Here $\mathrm{Unif}([a,b])$ denotes the continuous uniform distribution over the interval $[a,b]$.

These transforms give us a means of simulating monotone Markov processes with arbitrary coordinate-wise univariate marginal distributions at fixed time $t = 1$. Suppose the distribution G arises as the marginal of a monotone-increasing

[2] Recall that we write $\bar{G} \stackrel{\text{def}}{=} 1 - G$.

scalar stochastic *process* $\{\mathsf{G}(t)\}_t$ such that $\mathsf{G}(1) \sim G$. Then, processes

$$\{q^{G,F}(\mathsf{G}(t))\}_t \tag{3.10}$$

$$\{q^{\bar{G},F}(\mathsf{G}(t))\}_t \tag{3.11}$$

are, respectively, monotonic increasing and decreasing stochastic processes with the marginal distribution $F^{-1}(G(\mathsf{G}(1))) \overset{\mathrm{D}}{=} F^{-1}(\bar{G}(\mathsf{G}(1))) \sim F$. In problems where we have access to both F^{-1}, and G and where evaluating their composition is sufficiently cheap, these transforms comprise the workhorse tool to impose the desired marginal distribution. Note however, that generating random variates by the quantile transform method is often more expensive, computationally, than simulating directly from the target distribution where the latter is possible.

A common pattern applies in the case that we have a rare event of the form $\mathcal{L} \overset{\mathrm{def}}{=} \{S(\mathsf{X}) \leq \kappa\}$ for some quasi-monotone importance function $S : \mathbb{R}^d \to \mathbb{R}$ and X coordinate-wise independent, $\mathsf{X} \sim F_1 \times F_2 \times \cdots \times F_d$. Then, it is often possible to set $d = D$ and directly obtain a function $\boldsymbol{q} : \mathbb{R}^D \to \mathbb{R}^D$ such that $S_g = S \circ q$ is quasi-monotone. For example if S is itself quasi-monotone increasing, then we can construct $\boldsymbol{q}$ as

$$\boldsymbol{q}_{\mathrm{inc}} : \begin{pmatrix} g_1 \\ \vdots \\ g_d \end{pmatrix} \mapsto \begin{pmatrix} q^{E,F_1}(g_1) \\ \vdots \\ q^{E,F_d}(g_d) \end{pmatrix}. \tag{3.12}$$

Now $\{\mathsf{G}\}_t \sim \mathrm{GammaProc}(1,1)^{\times D}$ we have $\boldsymbol{q}_{\mathrm{inc}}(\mathsf{G}(1)) \overset{\mathrm{D}}{=} \mathsf{X}$ and thus $S_g = S \circ \boldsymbol{q}$ is an importance function for a left quasi-monotone problem as per Definition 3.1.

In a more general setting, suppose S is not quasi-monotone increasing in all components, but that the argument may be partitioned into two parts $\boldsymbol{x} = [\boldsymbol{x}^{\mathrm{inc}}; \boldsymbol{x}^{\mathrm{dec}}]$ where $\boldsymbol{x}^{\mathrm{inc}} = [x_1, \ldots, x_k]$ and $\boldsymbol{x}^{\mathrm{inc}} = [x_{k+1}, \ldots, x_d]$ such that S is quasi-monotone increasing with respect to $\boldsymbol{x}^{\mathrm{inc}}$ and decreasing in $\boldsymbol{x}^{\mathrm{dec}}$. Then, we can still find $\boldsymbol{q}$ such that $S_g = S \circ \boldsymbol{q}$ is quasi-monotone. We can create, say, a left-quasi-monotone problem by generating marginal distributions through respectively

increasing (3.4) and decreasing (3.5) quantile transforms.

$$\boldsymbol{q}_{\text{mixed}} : \left(g_1, \ \dots \ , g_d\right) \mapsto \begin{pmatrix} q^{E,F_1}(g_1) \\ \vdots \\ q^{E,F_k}(g_k) \\ q^{E,\bar{F}_{k+1}}(g_{k+1}) \\ \vdots \\ q^{E,\bar{F}_d}(g_d) \end{pmatrix}. \tag{3.13}$$

In such problems the $\boldsymbol{q}$ function may often serve also as the recovery function ρ. We use (3.12) and (3.13) often enough that where there is no ambiguity we refer to, e.g. "the" $\boldsymbol{q}_{\text{mixed}}$ for a given problem.

3.1.3 Subordinators in Quasi-monotone problems

Given the generality of quantile transforms we can afford to restrict the family of monotone latent processes G to a simple special case: processes with stationary and independent increments. This family of Lévy processes is known as the *subordinators*. We have already employed them in Example 2.8. The subordinators are described in greater depth in Appendix B, and exhaustively in e.g. Bertoin (1996), Kyprianou (2014), and Sato (1999). For the purpose of quasi-monotone splitting their crucial property is that calculating increment distributions at arbitrary times is easy by construction. Here we summarise the essential characteristics:

Definition 3.4 (Subordinator)**.** An $\mathbb{R}$-valued subordinator $\{G(t)\}_{t\in[0,t_{\max}]}\}_{t\in[0,t_{\max}]}$ is a Lévy process indexed by t and possessing the following qualities:

1. $G(t) - G(s)$ is independent of $G(u)$ for any $u < s < t$. (Independent increments.)
2. $G(s+t) - G(s)$ has the same distribution as $G(t) - G(0)$ for any $s, t > 0$. (Stationary increments.)
3. $G(s) \to G(t)$ in probability as $s \to t$. (Continuity in probability.)
4. $\mathbb{P}[G(t) - G(s) \geq 0] = \mathbb{I}\{t \geq s\}$. (Non-negative increments.)

A d-dimensional subordinator is an $\mathbb{R}^d$-valued stochastic process such that each coordinate is an independent scalar subordinator.

We further assume that the Lévy processes we use as latent processes have no deterministic positive linear *drift* term. Thus

$$\forall \varepsilon > 0, \mathbb{P}[\mathsf{G}(t+s) - \mathsf{G}(t) < \varepsilon] > 0. \tag{3.14}$$

An archetypal example of a subordinator is the gamma process, which we employ in the majority of our examples. We write $\text{GammaProc}(t; \alpha, \lambda)$ for associated distribution. Realizations of the process are shown in Figure 3.1 for various parameters. The salient feature for the current purpose is that $\{\mathsf{G}(t)\}_{t\in[0,\infty]} \sim \text{GammaProc}(t; \alpha, \lambda) \Rightarrow \mathsf{G}(t; \alpha, \lambda) - \mathsf{G}(s; \alpha, \lambda) \sim \text{Gamma}(\alpha(t-s), \lambda)$. These processes are described in detail in Section B.2.

Since its marginal distribution is absolutely continuous with respect to the Lebesgue measure on the positive real line, it is amenable to the quantile transforms (Definition 3.3). For example, if we choose $\{\mathsf{G}(t)\}_t \sim \text{GammaProc}(1, 1)$ it happens that the marginal distribution at time $t_{\max} = 1$ has the particularly simple form $\mathsf{G}(1) \sim \text{Exp}(1)$. Its CDF is thus

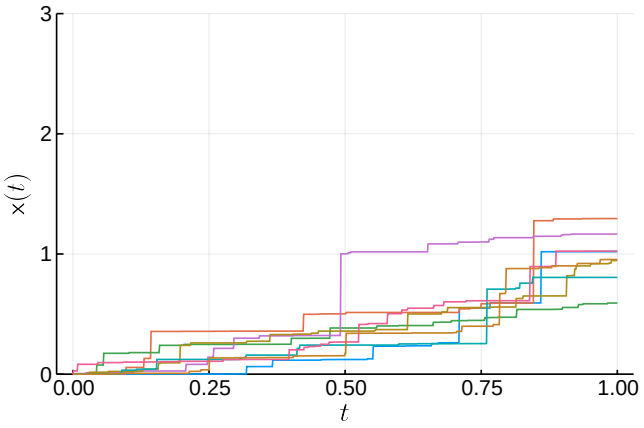
$$E(x) = 1 - \exp(-x). \tag{3.15}$$

Using the inverse lookup method we can construct arbitrary univariate marginal distributions to construct, respectively, monotone increasing and decreasing exponential maps

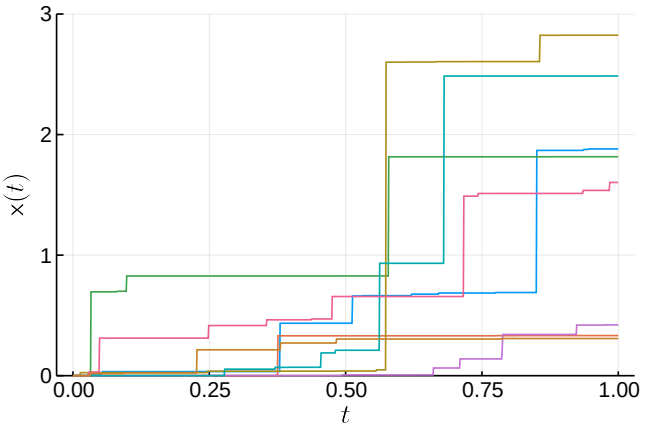
$$q^{E,F}(g) = F^{-1}(1 - \exp(-g)) \tag{3.16}$$
$$q^{E,\bar{F}}(g) = F^{-1}(\exp(-g)). \tag{3.17}$$

These transforms justify a default choice of $\mathsf{G} \sim \text{GammaProc}(1, 1)^{\times d}$ as our latent process, since this is in practice flexible enough for many marginals of interest. By a similar reasoning we set $t_{\max} = 1$ without loss of generality, since we can always rescale the time axis.

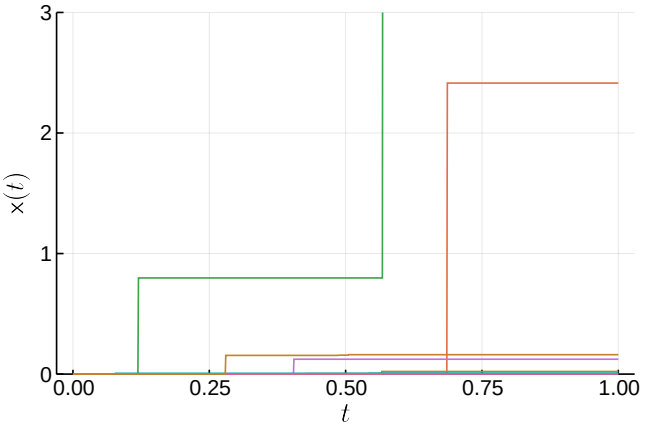
Although the inverse CDF constructions (3.4) and (3.5) can in principle generate any univariate marginal, in practice, CDFs may be computationally expensive or numerically unstable to invert. Thus it is useful to consider a broader class of latent subordinators than the gamma process alone. For example, tail probabilities



(a) $\lambda = 0.2$



(b) $\lambda = 1$



(c) $\lambda = 5$

Figure 3.1: Independent realizations of a gamma process G with $\mathbb{E}[G(1)] = 1$.

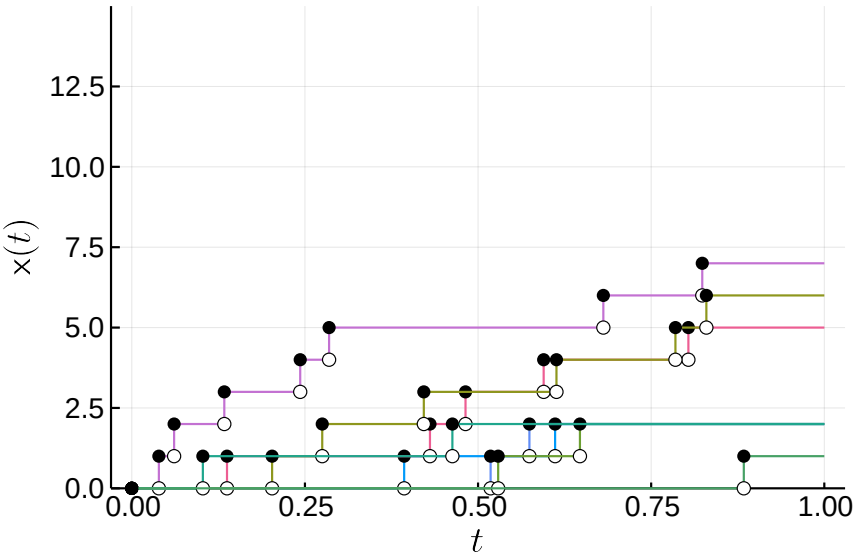


Figure 3.2: Independent realizations of a Poisson process G with $\mathbb{E}[G(1)] = 5$.

involving discrete marginals such as the Poisson are of interest, and inverting the Poisson CDF is not computationally efficient in the right tail of the distribution. Generally, if a problem directly involves the values of a subordinator distribution it may be easier to simulate its values directly, i.e., using a Poisson process, rather than by transformation of a gamma process. Such a case where it is more convenient to simulate the desired marginal directly is given in Subsection 3.3.3, where the variable in question has a Poisson marginal. Poisson processes are also subordinators, but with a Poisson, rather than gamma, increment distribution. Some representatives of the paths of such processes are shown in Figure 3.2, and a thorough introduction is given in Section B.3. More generally we might find other subordinators are convenient: compound Poisson processes, for example. Such elaborations are not necessary for the current purposes.

3.1.4 An example splitter

Example 3.1 (Gaussian Tail)**.** We begin with a test problem with an analytic solution against which we can cross-check our implementation. To this end, we

demonstrate a rare-event estimation problem involving a target set from a truncated Gaussian distribution. We wish to estimate the tail probability $\mathbb{P}[\boldsymbol{\beta}^\top \mathsf{X} > \kappa]$, where $\mathsf{X} = [\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_d]^\top \sim \mathcal{N}(\mathbf{0}, \mathrm{V})$. Here $\boldsymbol{\beta}$ is a vector of positive weights and κ is some arbitrary real. The target event is thus $\mathcal{L}_\kappa = \{\boldsymbol{\beta}^\top \mathsf{X} > \kappa\}$. When κ is large, this becomes a rare event problem.

We proceed in two stages. First, we stipulate a latent process comprising concatenated independent gamma processes, $\mathsf{G}_i \sim \mathrm{GammaProc}(1,1)^{\times d}$. Applying (3.17) to each component, $\mathsf{Z}_i(t) \stackrel{\text{def}}{=} \mathcal{N}^{-1}(\exp(-\mathsf{G}_i(t))$, we have constructed an intermediate vector-valued random process $\{\mathsf{Z}(t)\}_t$ such that it has a standard normal distribution at $t = 1$, i.e., $\mathsf{Z}(1) \sim \mathcal{N}(\mathbf{0}, \mathrm{I})$. (Here $\mathcal{N}^{-1}$ denotes the inverse complementary CDF of a standard normal.) To attain a normal vector with the desired covariance V from a standard Gaussian we can use the usual trick — given a standard d-dimensional normal variate Z, and a lower triangular matrix M such that $\mathrm{M}^\top \mathrm{M} = \mathrm{V}$, for strictly positive definite V, the linear transformation thereof, $\mathsf{X} = \mathrm{M}\mathsf{Z} \sim \mathcal{N}(\mathbf{0}, \mathrm{V})$ has the desired distribution. Thus we have that $\boldsymbol{q}(\mathsf{X}(1)) \sim \mathcal{N}(\mathbf{0}, \mathrm{V})$, where $\boldsymbol{q} \stackrel{\text{def}}{=} \mathrm{M}\mathcal{N}^{-1}(\exp(\mathsf{G}(1))$. Here $\boldsymbol{g} \mapsto \mathcal{N}^{-1}(\exp(\boldsymbol{g}))$ is applied coordinate-wise. The mapping of the level is the identity, $\alpha(\kappa) = \kappa$. Recovering a conditional sample in this problem is easy, with recovery function $\rho \equiv q$. The acceptance function becomes $S_g : \boldsymbol{g} \mapsto \boldsymbol{\beta}\boldsymbol{q}(\boldsymbol{g})$.

It remains to check the (right) quasi-monotone problem conditions (Definition 3.2,) which means that S_g must be quasi-monotone decreasing. For each $i = 1, \ldots, d$ the coordinate-wise map $g_i \mapsto \mathcal{N}^{-1}(\exp(-g))$ is decreasing. Thus we require that $\forall i$, $\mathrm{M}\boldsymbol{x}' \geq \mathrm{M}\boldsymbol{x}$ where $\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_d]^\top$ and $\boldsymbol{x}' = [x_1, \ldots, x_i', \ldots, x_d]^\top$ is a perturbed copy with $x_i' \geq x_i$. Equivalently, we require $\mathrm{M}(\boldsymbol{x}' - \boldsymbol{x}) \geq 0$ coordinate-wise, and thus $\mathrm{M}[0, \ldots, (x_i' - x_i), \ldots, 0]^\top \geq 0$, which implies all the elements of M must be non-negative. This problem is thus amenable to our method for all $\mathrm{V} = \mathrm{M}^\top \mathrm{M}$ for coordinate-wise non-negative lower-triangular M. As an aside, we note that a similar argument allows us to simulate more generally from Gaussian copulas with a positive covariance decomposition and arbitrary marginal distributions.

In this contrived case we can recover the true target probability in terms of the

standard univariate normal CCDF since

$$\begin{aligned}
\mathbb{P}[\mathcal{L}] &= \mathbb{P}[\{\boldsymbol{\beta}^\top \boldsymbol{X} > \kappa\}] && (3.18)\\
&= \mathbb{P}[\{\boldsymbol{\beta}^\top \mathrm{M} \boldsymbol{Z} > \kappa\}] && (3.19)\\
&= \mathbb{P}\left[\left\{Z_1 > \frac{\kappa}{\|\mathrm{M}^\top \boldsymbol{\beta}\|_2}\right\}\right] \qquad \text{with } Z_1 \sim \mathcal{N}(0,1) && (3.20)\\
&= \bar{\mathcal{N}}\left(\frac{\kappa}{\|\mathrm{M}^\top \boldsymbol{\beta}\|_2}; 0, 1\right). && (3.21)
\end{aligned}$$

We explore this model numerically. We fix parameters, choosing a two-dimensional model which we may easily plot, where

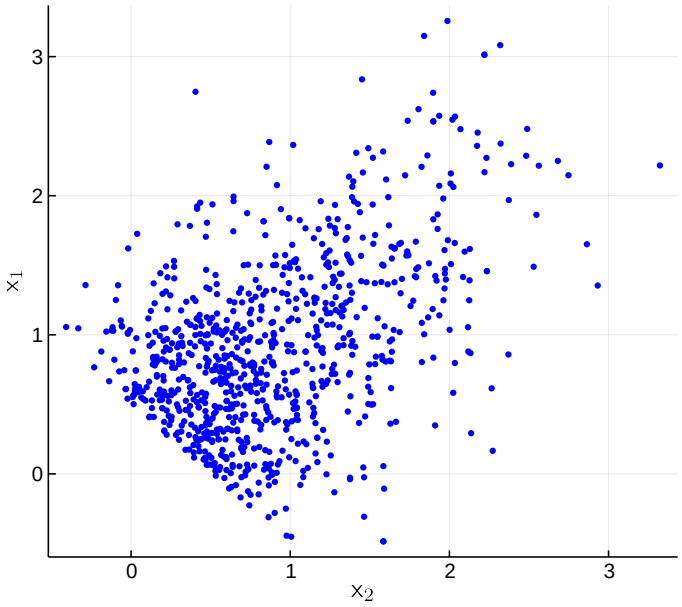
$$\boldsymbol{X} \sim \mathcal{N}\left(0, \begin{pmatrix} 1 & 0.8 \\ 0.8 & 1 \end{pmatrix}\right) \text{ and} \tag{3.22}$$

$$\boldsymbol{\beta} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{3.23}$$

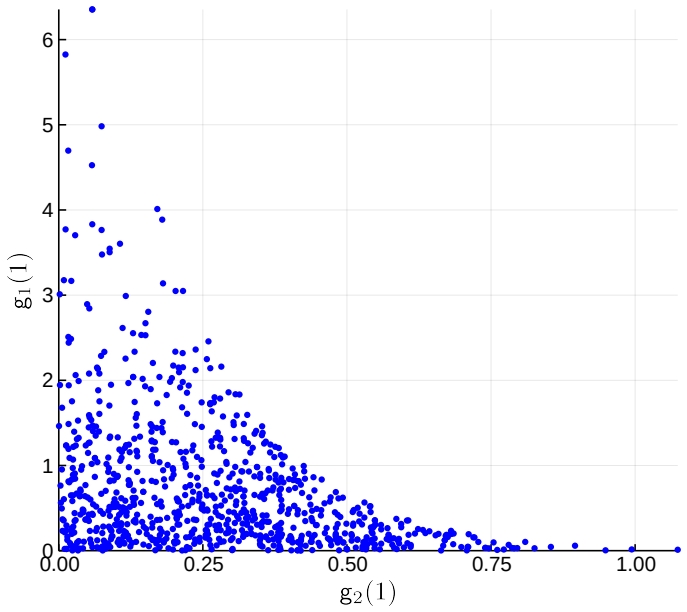
The simulated results are shown in Table 3.1, and plotted in Figure 3.3, and contrasted with the true value of the relative error (3.21). The results do not contradict our supposition that this provides an estimator whose value hews closely to the estimand.

Table 3.1: Probability estimates in the 2-dimensional Gaussian right-tail problem with $\tilde{n} = 10^4$ particles.

κ	ℓ	$\hat{\ell}$	r̂e(%)
1	0.299	0.302	0.852
2	0.146	0.146	0.0619
3	0.0569	0.0567	0.431
4	0.0175	0.0174	0.562
5	0.0042	0.00412	2.04
6	0.000783	0.000792	1.16



(a) Ambient space values, $X = \rho(G(1))|\mathcal{L}_\kappa$



(b) Latent process, $G(1)|\mathcal{L}_\kappa$

Figure 3.3: 1000 realizations generated by splitting simulation for the right Gaussian tail model.

3.2 Intermediate target event selection

In this section we develop a heuristic algorithm for ensuring low-variance estimates by choosing optimal times. This is aligned with the version developed for Ben Rached et al. (2020). We return to a more extensive treatment of the topic of efficient optimal intermediate target event selection in Chapter 4.

Our criterion is that the ideal uniform splitting level relation (2.75) holds (approximately) between splitting levels, i.e., that the conditional survival probabilities $p^{(m)}$ are constant, which have argued leads to large-effort asymptotically optimal estimates, at least in idealized problems. Writing out the dependence on intermediate target sets explicitly, this requires choosing $\mathcal{L}^{(m)}$ to approximately satisfy, for all m,

$$p^{(m)} \stackrel{\text{def}}{=} \mathbb{P}[\mathcal{L}^{(m)} \mid \mathcal{L}^{((m-1))}] = \frac{\mathbb{P}[\mathcal{L}^{(m)}]}{\mathbb{P}[\mathcal{L}^{((m-1))}]} = \check{p} \tag{3.24}$$

for some fixed desired conditional survival probability $\check{p}$. In the quasi-monotone splitting setting, we are given already a natural choice for S_g and we have already demonstrated that *kappa* should be held fixed. The intermediate target events and thus the conditional survival probabilities (3.24) are uniquely determined for fixed model parameters by the splitting instants $t_m, m = 1, \ldots, M$. We thus consider a *time selection* procedure.

Accordingly we model the *lifetime distribution*, $\mathsf{T}(\kappa) \sim T(\cdot; \kappa)$, i.e., the distribution of the random variable

$$\mathsf{T}(\kappa) \stackrel{\text{def}}{=} \inf\{t : S_g(\mathsf{G}(t)) \geq \kappa\}. \tag{3.25}$$

Where we hold κ constant we suppress it and simply discuss $\mathsf{T} \sim T$. This is by construction a non-negative real random variable which measures the exit time of a particle from the target event in the quasi-monotone splitting estimator.

The distribution of T gives us some insight into the behaviour of the algorithm we have developed. Firstly, we note that T is of unbounded support. Since the latent processes have no deterministic drift it follows from (3.14), the quasi-

monotonicity of S_g and (3.25) that, conditional on $S_g(G(0)) < \kappa]$,

$$\mathbb{P}[T > M] > 0, \quad \forall M > 0 \tag{3.26}$$

That is, the lifetime distribution is unbounded above and $\mathrm{supp}(T) = [0, \infty)$.

The nesting condition (2.46) for the intermediate target events under quasi-monotonicity is then

$$\{T > t_m\} \supseteq \{T > t_{m+1}\}. \tag{3.27}$$

With regard to T we may rewrite the criterion (3.24) using (3.25) as

$$\check{p} = \mathbb{P}\left[\mathcal{L}^{(m)} \mid \mathcal{L}^{(m-1)}\right] = \frac{\mathbb{P}\left[S_g(G(t_m)) \leq \kappa\right]}{\mathbb{P}\left[S_g(G(t_{m-1})) \leq \kappa\right]} = \frac{\mathbb{P}[T > t_m]}{\mathbb{P}[T > t_{m-1}]}. \tag{3.28}$$

We note that $\ell = \mathbb{P}[T > t] = 1 - T(t) = \bar{T}(t)$. Recursively applying (3.28), we see the optimal times $t_1, \ldots, t_M$ must satisfy

$$t_m = \bar{T}^{-1}(\check{p}^m), \tag{3.29}$$

Of course, if we knew $\bar{T}$ we would also know $\ell = \bar{T}(1)$, which is in general not the case. Our solution to this difficulty is to dedicate a small proportion of the simulation effort to a pilot run that guides the main simulation effort in the hope that in combination these methods can be more efficient. Investigating effort allocation to ensure this is the topic of the sequel, Chapter 4. We introduce here the convention that we distinguish the parameters of the pilot run from those of the main run by marking the pilot parameters with a prime, e.g. t'_k is a pilot run splitting time.

The first step in the piloted quasi-monotone method applies an adaptive pilot algorithm (Algorithm 3.2) which we explain in detail momentarily, in order to determine a step count K and a set of intermediate splitting times $\{t'_k\}_{k=1}^{K-1}$ with $t_K = 1$, and estimates of $\{\hat{\bar{T}}(t_k)\}_{k=1}^{K}$. From the output of a pilot run, we construct point estimates $\hat{T}(t'_k)$ at pilot times t'_k, $k = 1, \ldots, K$, and interpolate between these using a piecewise linear interpolant. We use a plug-in estimate of $\bar{T}(t_k) = \ell'^{(k)}$ which we obtain as a side-effect of the quasi-monotone splitting estimator (i.e., in Line 16 of Algorithm 3.1). This once again invokes the idealization of the splitting

problem that we used in (2.65), in that we ignore any dependence upon the state of the particles in the conditional survival probabilities, and further, ignore the dependence between particles. We impose a particular form upon the CCDF, in particular estimating that CCDF function $\bar{T}(\cdot)$ by linearly interpolating between these point estimates. The resulting CCDF estimator is a linear spline using knot times $0, t'_1, t'_2, \ldots, t'_K = 1$ and corresponding values $1, \hat{\bar{T}}(t'_1), \ldots, \hat{\bar{T}}(t_K)$ (marked as green points in the diagram). The resulting pilot interpolant is shown as the blue curve. By construction this interpolant is over $[0, 1]$. We require $t_M = 1$, so we choose

$$M = 1 + \max_{m \in \mathbb{N}} \hat{\bar{T}}^{-1}(\check{p}^m) < 1 = \lceil \log \hat{\ell}' / \log \check{p} \rceil. \tag{3.30}$$

Finally, we estimate remaining times by the plug-in method, inverting $\hat{\bar{T}}$ as per (3.29), giving $t_m = \bar{T}^{-1}(\check{p}^m)$, $m = 1, \ldots, M-1$. The resulting times satisfy (3.24) with regard to the estimated CCDF $\hat{\bar{T}}$ except for the final time increment, for which $p^{(M)} \leq \check{p}$. These estimated ideal splitting times $t_m, m = 1, \ldots, M$ are shown with orange points. These splitting times and M values are themselves, properly speaking, random estimators which would normally be written $\hat{t}_m$ and $\hat{M}$. We suppress the hatted estimator notation, however, to reduce clutter. This method is depicted in Figure 3.4.[3]

The *adaptive* pilot algorithm is a heuristic method to choose ideal times $\{t'_k\}_{k=1}^{\bar{K}-1}$ in an online search. We conduct an approximate random binary search, starting from $t_0 = 0$ and iteratively finding the next time $t_{k+1} = t_k + \delta$ via a trial and error search over δ. If δ is too large, then the proportion of states that survive is too small (less than p^*) and so we then try a smaller $\delta \leftarrow \delta/2$ until the number of surviving states is at least p^*. As a consequence of the nesting property of splitting, the smaller is δ, the larger is the expected number of surviving states. Pseudo code of the algorithm is given in Algorithm 3.2. The choice for $\tilde{n}'$ in the pilot should be such that the cost of the pilot is a small fraction of the cost of the main splitting run and thus insignificant in the eventual efficiency comparison. We return to what in 'insignificant' means in practice in Chapter 4.

[3] We observe parenthetically that the "bumpy" appearance of the CCDF interpolant on the log scale is suggestive that the interpolating curve is could possibly be improved by conducting interpolation in the log domain. Indeed, that is one one of the extensions explored in Chapter 4.

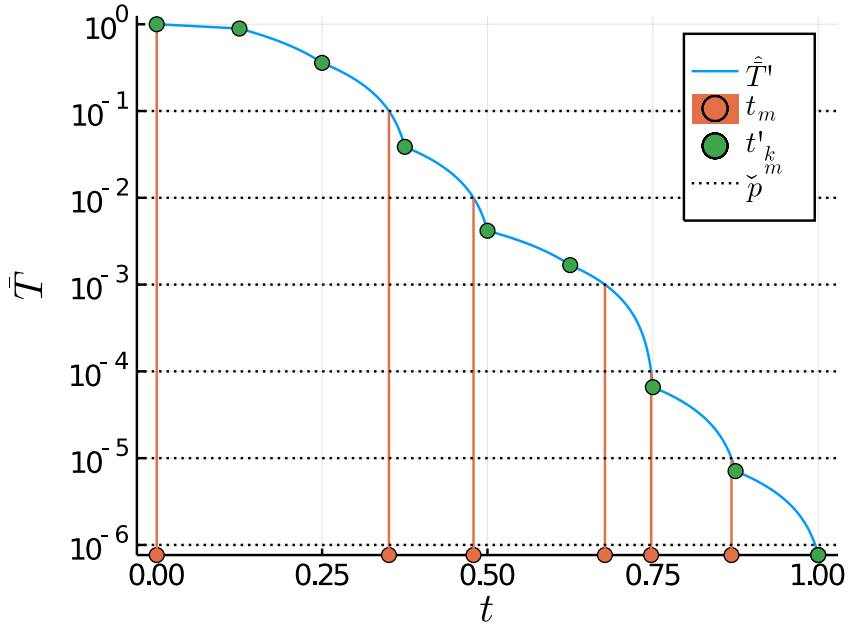


Figure 3.4: Stylized depiction of method of time selection by linear CCDF interpolation. $\check{p} = 0.1$. Note that the vertical log scale converts the linear interpolants into piecewise convex curves.

Algorithm 3.2 Adaptive pilot run to estimate $\{\hat{\tilde{T}}(t_i)\}$.

Require: Population count $\tilde{n}'$.
Require: Initial time increment δ
Require: Probability estimate lower bound p^*.
Ensure: $\{\hat{\tilde{T}}(t'_k); k = 1, 2, \dots\}$, pointwise estimates of the complementary life distribution CDF.

loop
  $t'_1 \leftarrow \min\{\delta, 1\}$
  Simulate forward: $\tilde{\mathcal{G}}' \leftarrow \{\boldsymbol{\xi}'_{(i)}(t'_1) \sim G(\cdot; \delta : i \in 1, \dots, \tilde{n}\}$
  Prune: $\mathcal{G}' \leftarrow \left\{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\tilde{\mathcal{G}}) \text{ if } S_g(\boldsymbol{\xi}_{(i)}) \leq \kappa, \right\}$
  if $\frac{|\mathcal{G}|}{|\tilde{\mathcal{G}}|} \geq p^*$ **then**
    EXIT the loop
  else
    $\delta \leftarrow \delta/2$
$\hat{p}^{(1)} \leftarrow \frac{|\mathcal{G}|}{|\tilde{\mathcal{G}}|}$
$\hat{\ell}^{(1)} \leftarrow \hat{p}^{(1)}$
$t_1 \leftarrow t'_1$
$\hat{\tilde{T}}(t'_1) \leftarrow \hat{\ell}^{(1)}$
Split states: $\mathcal{G} \leftarrow \tilde{n}$ samples from $\mathcal{G}$ drawn uniformly with replacement.
$\delta \leftarrow 2\delta$ (try increasing the size of δ)
$k \leftarrow 2$
while $t_{k-1} < 1$ **do**
  loop
    $t'_k \leftarrow \min\{t_{k-1} + \delta, 1\}$
    Simulate forward: $\tilde{\mathcal{G}}' \leftarrow \{\boldsymbol{\xi}'_{(i)}(t'_k) \sim G(\cdot; t_{k-1} + \delta \mid \boldsymbol{\xi}_{(k)}(t_{k-1}) : i \in \mathcal{I}(\mathcal{G})\}$
    Prune: $\mathcal{G}' \leftarrow \left\{\boldsymbol{\xi}_{(i)} \text{ for } i \in \mathcal{I}(\tilde{\mathcal{G}}) \text{ if } S_g(\boldsymbol{\xi}_{(i)}) \leq \kappa, \right\}$
    if $\frac{|\mathcal{G}|}{|\tilde{\mathcal{G}}|} \geq p^*$ **then**
      EXIT the loop
    else
      $\delta \leftarrow \delta/2$
  $\hat{p}^{(k)} \leftarrow \frac{|\mathcal{G}|}{|\tilde{\mathcal{G}}|}$
  $\hat{\ell}^{(k)} \leftarrow \hat{p}^{(k)}\hat{\ell}^{(k-1)}$
  $t_k \leftarrow t'_k$
  $\hat{\tilde{T}}(t_k) \leftarrow \hat{\ell}^{(k)}$
  Split states: $\mathcal{G} \leftarrow \tilde{n}$ samples from $\mathcal{G}$ drawn uniformly with replacement.
  $k \leftarrow k + 1$
  $\delta \leftarrow 2\delta$ (try increasing the size of δ)
return $K \leftarrow k$ and $\{t_i\}_{k=1}^K$ and $\{\hat{\tilde{T}}(t_i)\}$

The adaptive pilot algorithm has, empirically, acceptable performance, but it poses certain difficulties for analysis. There are no clear guidelines on how to choose δ, $\tilde{n}$ or p^*. Further, the relationship between the values of these parameters and the simulation effort in any given problem is opaque, which complicates effort-normalized variance analysis. With these caveats, we retain this method for the remainder of the chapter, revisiting it in the sequel. We use empirical point estimates of the WNRV for approximate comparison of these methods. For the moment we set $p^* = 0.1$, $\delta = 0.1$, $\tilde{n}' = 200$ and $\check{p} = 0.2$.

3.3 Some quasi-monotone splitting probability estimators

It turns out that a number of practical problems of industrial utility are quasi-monotone. In this section we explore a selection of such estimation problems from multiple fields, with a particular emphasis on problems from wireless network reliability estimation. We concern ourselves mostly with rare-event truncated probability estimates of $\mathbb{P}[\mathcal{L}]$. At the end there is with some tail-conditional sampling $\mathbb{E}[\phi(X \mid \mathcal{L}]$.

The rare events of interest here are drawn from heavy-tailed families, including log-normal, Weibull, and Generalized Pareto random variates. Log-normal random variate problems arise in finance and biology, in for example, pricing of Asian options and modelling bacterial growth (Botev, Salomone, and Mackinlay 2019; Limpert, Stahel, and Abbt 2001). In reliability engineering, generalized Pareto and Weibull variates are also important (Simon and Alouini 2005).

The importance functions here have simple forms. For instance, *signal-to-noise-ratio* (SNR) and *outage probability* (OP) calculations may be expressed as sums of random variables or partial sums of ordered random variables (Ben Rached et al. 2016; Ben Rached et al. 2018b). In the presence of co-channel interferences and noise we are concerned with the *signal-to-interference-plus-noise ratio* (SINR), which is the ratio of the desired power signal and the sum of interfering power signals plus noise (Ben Rached et al. 2017; Botev, Salomone, and Mackinlay 2019).

In addition to these applications, in which the considered RVs are typically con-

tinuous, there are demands both in wireless communication applications (Bashir and Alouini 2020) and physics (Tan, Lu, and Xia 2018) for estimands defined by sums of discrete RVs. Of particular interest is the probability that a weighted sum of independent Poisson RVs falls below a small threshold, which represents the probability of missed detection of alignment in free space optical communication systems. All the above applications can be handled as quasi-monotone problems. This catalogue is not exhaustive. Many further applications are straightforward extensions. We will discuss further classes of problems made tractable by this method after introducing some concrete examples.

For the examples in this selection, we introduce the structure of the problem, then stipulate how the design parameters G, S_g, α and ρ are to be chosen for a given problem to attain a quasi-monotone structure matching Definition 3.1. In order to estimate the variance of estimators we run the algorithm for $R = 200$ replications and estimate performance statistics numerically from these replicated estimates using (2.33)-(2.38). The number of samples per level is for now fixed at $\tilde{n} = 3000$.

3.3.1 A partial sum problem

An important class of problems in, for example, telecommunications, is the *partial sum of order statistics*. These problems arise in, for example, reliability estimates in wireless networking outage probabilities with certain transmission strategies (Ben Rached et al. 2016; Ben Rached et al. 2018b). In these systems, we need to know exceedance probabilities for partial sums of order statistics of independent RVs. Concretely, we wish to find $\ell = \mathbb{P}[\mathcal{L}]$ for $\mathcal{L} = \{S(\mathsf{X}) \leq \kappa\}$ with

$$S(\mathsf{X}) \stackrel{\text{def}}{=} \sum_{i=1}^{d} \mathsf{X}^{[i]}. \tag{3.31}$$

and some $1 \leq d \leq D$. Here $\mathsf{X}^{[1]}, \cdots, \mathsf{X}^{[D]}$ are the order statistics (i.e., the variables sorted by value) of the ambient RVs $\mathsf{X}_1 \sim F_1, \mathsf{X}_2 \sim F_2, \ldots, \mathsf{X}_D \sim F_D$. The sort is in decreasing order, $\mathsf{X}^{[1]} \geq \mathsf{X}^{[2]} \geq \cdots \geq \mathsf{X}^{[D]}$. Closed-form results are available in certain restricted cases, e.g. when $\mathsf{X}_1, \cdots, \mathsf{X}_D$ are exponential, gamma, or generalized gamma distributed RVs (Bithas, Sagias, and Mathiopoulos 2007;

Nam, Alouini, and Yang 2010; Nam, Ko, and Alouini 2017), under additional restrictions upon the parameters of the generating process. Specialized simulation methods have been recently proposed in some recent works (Ben Rached et al. 2018a; Ben Rached et al. 2018b), in which the RVs $\mathsf{X}_1, \cdots, \mathsf{X}_D$ are generalized gamma or log-normal variates, and these are, to our knowledge, the most efficient available estimators for this particular problem where applicable.

This has introduced the version of the problem where we consider the largest-d variates from the list of D variables, but generalizing is straightforward: estimators for the version where we consider instead the *smallest-d* variables, and for the right handed version with $\mathcal{L}_\kappa = \{S(\mathsf{X}) \geq \kappa\}$ require no special treatment.

The application of quasi-monotone splitting to this problem is immediate. Observing that sorting the list of random variables preserves quasi-monotonicity, we take latent process $\mathsf{G} \sim \mathrm{GammaProc}(1,1)^{\times D}$ and use quantile transforms to construct the desired ambient RVs, $\mathsf{X}_i \overset{\mathrm{D}}{=} q^{E,F_i}(\mathsf{G}_i(1))$. We take $S_g : \boldsymbol{g} \mapsto \sum_{i=1}^{d} q^{E,F_i}(g_i)^{[i]}$, where $q^{E,F_i}(g_i)^{[i]}$ means the ith element in the vector obtained by sorting $[q^{E,F_1}(g_1), q^{E,F_2}(g_2), \ldots, q^{E,F_D}(g_D)]$ into decreasing order. The recovery function ρ is similar but returns only the first d coordinates of the sorted vector. The result is a left quasi-monotone problem. Unlike the existing Monte Carlo methods, we have introduced no assumptions on the distribution of the vectors, and they may have different distributional parameters or come from different families. As long as the required CDF inversion is feasible, the overall method remains feasible.

We perform numerical simulations of estimators where the component random variables have the Weibull and log-normal distributions. The Weibull CDF is given as follows

$$F_{\mathrm{Weibull}}(x;\alpha,\lambda) = 1 - \mathrm{e}^{-(x/\lambda)^\alpha} x \geq 0 \; x \; > 0, \tag{3.32}$$

with $\alpha, \lambda > 0$ denoting the shape and the scale parameters respectively. Weibull distributions are included in the generalized gamma family, and thus amenable to the specialized estimators of Ben Rached et al. (2018a) and Ben Rached et al. (2018b). A log-normal distribution is the distribution of the exponential transform

of a normal variate,

$$X \sim \text{LogNormal}(\mu, \sigma^2) \Rightarrow \log X \sim \mathcal{N}(\mu, \sigma^2). \tag{3.33}$$

We implement quasi-monotone splitting for partial sum problems for variables involving both of these, and also side-by-side implementation with the best available alternative Monte Carlo methods for comparison.

The results in Table 3.2 and Table 3.3 show quasi-monotone splitting for Weibull variates with two efficient estimators proposed in Ben Rached et al. (2018b), over different values of D and d. The two alternative methods are a variance-reduced IS and a problem-specific "conditionalized Monte Carlo". They both leverage the structures of the underlying distributions (generalized gamma, and in particular, the Weibull distribution) to attain efficient estimates — bounded relative error in some cases. As such these dominate our mere near-logarithmic efficiency. Indeed, quasi-monotone splitting does not outperform these alternatives, having a strictly and substantially inferior WNRV over all of the κ values. In contrast, when our random variates are log-normally distributed and the same tricks are inapplicable, the quasi-monotone method can estimate the target values generically and often provides the best performance. Simulation results in Table 3.4 and Table 3.5 demonstrate superior performance for quasi-monotone splitting for certain choices κ. Concretely, quasi-monotone splitting achieves 10 times greater efficiency than the conditional MC estimator for the parameters of Table 3.4 when $\kappa = 0.15$. The quasi-monotone splitting method also required minimal user derivations — specifically, our simple derivation for this problem fits into a single paragraph, whereas the competing methods require extensive calculation.

Table 3.2: $\hat{\ell}$ for the partial sum of Weibull order statistics with $D = 8$, $d = 4$, $\alpha = 0.5$, $\xi = 1$.

	Quasi-monotone splitting			Universal IS estimator with $m = 5 \cdot 10^5$			Conditional MC estimator with $m = 5 \cdot 10^5$		
κ	$\hat{\ell}$	$\widehat{\text{re}}(\hat{\ell})\%$	$\widehat{\text{WNRV}}(\hat{\ell})$.	$\hat{\ell}$	$\widehat{\text{re}}(\hat{\ell})\%$	$\widehat{\text{WNRV}}(\hat{\ell})$	$\hat{\ell}$	$\widehat{\text{re}}(\hat{\ell})\%$	$\widehat{\text{WNRV}}(\hat{\ell})$
1	0.0029	0.61	$4.78 \cdot 10^{-4}$	0.0029	0.40	$7.68 \cdot 10^{-6}$	0.0029	0.12	$1.72 \cdot 10^{-5}$
0.5	$3.36 \cdot 10^{-4}$	0.94	$1.5 \cdot 10^{-3}$	$3.37 \cdot 10^{-4}$	0.49	$1.15 \cdot 10^{-5}$	$3.37 \cdot 10^{-4}$	0.13	$2.02 \cdot 10^{-5}$
0.1	$1.26 \cdot 10^{-6}$	1.36	$4.5 \cdot 10^{-3}$	$1.27 \cdot 10^{-6}$	0.66	$2.09 \cdot 10^{-5}$	$1.27 \cdot 10^{-6}$	0.15	$2.70 \cdot 10^{-5}$
0.05	$9.80 \cdot 10^{-8}$	1.51	$6.4 \cdot 10^{-3}$	$9.85 \cdot 10^{-8}$	0.71	$2.42 \cdot 10^{-5}$	$9.79 \cdot 10^{-8}$	0.16	$3.07 \cdot 10^{-5}$
0.01	$2.10 \cdot 10^{-10}$	1.90	$1.43 \cdot 10^{-2}$	$2.06 \cdot 10^{-10}$	0.80	$3.07 \cdot 10^{-5}$	$2.07 \cdot 10^{-10}$	0.17	$3.46 \cdot 10^{-5}$
0.005	$1.39 \cdot 10^{-11}$	2.05	$2.03 \cdot 10^{-2}$	$1.39 \cdot 10^{-11}$	0.81	$3.15 \cdot 10^{-5}$	$1.38 \cdot 10^{-11}$	0.17	$3.46 \cdot 10^{-5}$

Table 3.3: $\hat{\ell}$ for the partial sum of Weibull order statistics with $D = 8$, $d = 4$, $\alpha = 0.8$, $\xi = 1$.

	Quasi-monotone splitting			Universal IS estimator with $m = 5 \cdot 10^5$			Conditional MC estimatorwith $m = 5 \cdot 10^5$		
κ	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$.	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$
1.03	$3.38 \cdot 10^{-4}$	0.93	$1.5 \cdot 10^{-3}$	$3.41 \cdot 10^{-4}$	1.28	$9.99 \cdot 10^{-5}$	$3.37 \cdot 10^{-4}$	0.1	$1.28 \cdot 10^{-5}$
0.38	$1.31 \cdot 10^{-6}$	1.42	$5.1 \cdot 10^{-3}$	$1.29 \cdot 10^{-6}$	2.31	$3.20 \cdot 10^{-4}$	$1.31 \cdot 10^{-6}$	0.12	$1.74 \cdot 10^{-5}$
0.09	$2.09 \cdot 10^{-10}$	2.00	$1.71 \cdot 10^{-2}$	$2.22 \cdot 10^{-10}$	3.20	$6.24 \cdot 10^{-4}$	$2.10 \cdot 10^{-10}$	0.13	$2.06 \cdot 10^{-5}$
0.058	$1.36 \cdot 10^{-11}$	2.29	$2.90 \cdot 10^{-2}$	$1.33 \cdot 10^{-11}$	3.49	$7.43 \cdot 10^{-4}$	$1.35 \cdot 10^{-11}$	0.13	$2.04 \cdot 10^{-5}$

Table 3.4: $\hat{\ell}$ for the partial sum of log-normal order statistics with $D = 8$, $d = 4$, $\mu = 0$, $\sigma = 2$.

	Quasi-monotone splitting			Universal IS estimator with $m = 10^6$			Conditional MC estimator with $m = 10^6$		
κ	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$.	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$
1	$8.30 \cdot 10^{-5}$	1.00	$2.3 \cdot 10^{-3}$	$8.31 \cdot 10^{-5}$	0.68	$5.08 \cdot 10^{-5}$	$8.31 \cdot 10^{-5}$	0.34	$8.57 \cdot 10^{-4}$
0.5	$1.91 \cdot 10^{-6}$	1.35	$5.3 \cdot 10^{-3}$	$1.91 \cdot 10^{-6}$	1.27	$1.82 \cdot 10^{-4}$	$1.90 \cdot 10^{-6}$	0.99	$7.2 \cdot 10^{-3}$
0.3	$7.04 \cdot 10^{-8}$	1.63	$1.03 \cdot 10^{-2}$	$7.07 \cdot 10^{-8}$	2.11	$5.07 \cdot 10^{-4}$	$7.00 \cdot 10^{-8}$	2.10	$3.19 \cdot 10^{-2}$
0.15	$3.93 \cdot 10^{-10}$	2.12	$2.12 \cdot 10^{-2}$	$3.90 \cdot 10^{-10}$	4.37	$2.20 \cdot 10^{-3}$	$3.92 \cdot 10^{-10}$	5.41	$2.09 \cdot 10^{-1}$

Table 3.5: $\hat{\ell}$ for the partial sum of log-normal order statistics for with $D = 15$, $d = 15$, $\mu = 0$, $\sigma = 2$.

	Quasi-monotone splitting			Universal IS estimator with $M = 5 \cdot 10^7$		
κ	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$.	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$
3.4	$1.93 \cdot 10^{-6}$	1.98	$2.48 \cdot 10^{-2}$	$2.00 \cdot 10^{-6}$	0.94	$7.3 \cdot 10^{-3}$
2.3	$6.74 \cdot 10^{-8}$	3.04	$6.73 \cdot 10^{-2}$	$6.50 \cdot 10^{-8}$	2.50	$5.25 \cdot 10^{-2}$
1.39	$3.68 \cdot 10^{-10}$	3.44	$9.31 \cdot 10^{-2}$	$4.20 \cdot 10^{-10}$	9.58	$6.79 \cdot 10^{-1}$

3.3.2 A ratio problem

We consider now SINR problems, wherein we must evaluate the probability of rare tail events generated by ratios of non-negative random variables of the form

$$\mathbb{P}\left[\frac{X_1}{\sum_{i=2}^{d} X_i + E} \leq \kappa\right] \tag{3.34}$$

where X_1 represents the signal of interest, $X_2, \ldots, X_d$ interfering signals and E the noise. (Ben Rached et al. 2017; Botev, Salomone, and Mackinlay 2019). All RVs are independent, with densities given $X_i \sim F_i, E \sim F_e$. We may take this as a problem with importance function

$$S(\boldsymbol{x}) = \frac{x_1}{\sum_{i=2}^{d+1} x_i}. \tag{3.35}$$

Noting that the ambient importance function is increasing in X_1 and decreasing in all other RVs, we can find a quasi-monotone structure with respect to latent process vector $G \sim \text{GammaProc}(1,1)^{\times(d+1)}$ by using a combination of increasing (3.4) and decreasing (3.5) quantile transforms (i.e. we use the $S_g \overset{\text{def}}{=} S \circ \boldsymbol{q}_{\text{mixed}}$ structure).

$$S_g(G) \overset{\text{def}}{=} \frac{q^{E,F_1}(G_1(t))}{\sum_{i=2}^{d} q^{E,\bar{F}_i}(g)(G_i(t)) + q^{E,\bar{F}_e}(g)(G_{d+1}(t))}. \tag{3.36}$$

In this problem the best competing algorithm to our knowledge is that of Ben Rached et al. (2017), which applies to log-normally distributed RVs $X_1, \ldots X_1, E$. That algorithm leverages the observation that the target event may be transformed into a problem of exceedance probabilities for correlated log-normal RVs, and thus may be approached using importance sampling based on scaling the covariance matrix of the corresponding Gaussian vector. In the comparison we apply that log-normal IS algorithm with $n = 2 \cdot 10^6$ importance samples.

The result of the comparison is in Table 3.6. The values of the relative errors and WNRV show superiority in this case for quasi-monotone splitting attaining, e.g., a fivefold better estimated efficiency than the alternative when $\gamma = 0.001$. In this case, splitting is not only more efficient than the alternative but also more general, being applicable outside the log-normal distribution already — all we need do is change the CDFs used in the quantile transforms.

Table 3.6: $\hat{\ell}$ for $X_1/(\sum_{i=2}^{d} X_i + E)$ for log-normal variates with $d = 11$, $\mu_0 = 20$ dB, $\mu = 0$ dB $\sigma = 4$ dB, $\sigma_0 = 6$ dB, $E = -10$ dB.

	Quasi-monotone splitting			Variance scaling IS with $M = 2 \cdot 10^6$		
κ	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$.	$\hat{\ell}$	$\widehat{\mathrm{re}}(\hat{\ell})\%$	$\widehat{\mathrm{WNRV}}(\hat{\ell})$
0.02	$2.11 \cdot 10^{-5}$	1.41	$5.0 \cdot 10^{-3}$	$2.01 \cdot 10^{-5}$	2.70	$1.33 \cdot 10^{-2}$
0.003	$2.90 \cdot 10^{-8}$	2.03	$1.82 \cdot 10^{-2}$	$2.83 \cdot 10^{-8}$	7.30	$1.2 \cdot 10^{-1}$
0.001	$2.94 \cdot 10^{-10}$	2.40	$3.43 \cdot 10^{-2}$	$3.35 \cdot 10^{-10}$	11.45	$1.72 \cdot 10^{-1}$

3.3.3 Poisson sum problem

An advantage of the proposed splitting method is its ability to handle functionals of discrete RVs as well as continuous RVs. Let $X_1, X_2, \cdots, X_d$ be a sequence of independent Poisson RVs with rates $\lambda_1, \lambda_2, \cdots, \lambda_n$ and density

$$\mathbb{P}[X_i = k] = \lambda_i^k \exp(-\lambda_i)/k!, \quad k = 0, 1, 2, \ldots \tag{3.37}$$

We would like to estimate the probability of weighted sum of these variates being below a threshold κ, i.e.

$$S(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{i=1}^{d} w_i x_i < \kappa, \tag{3.38}$$

where w_i are non-negative weights.

A Poisson process may be used as the latent (and in fact, ambient) process in quasi-monotone splitting. This is because each Poisson variable X_i already has the same marginal distribution as the continuous-time Poisson process $\{X_i(t), t \geq 0\}$ with $X_i(0) = 0$ at time $t = 1$. Poisson processes are indeed already subordinators (Section B.3), so for this problem we take $S_g \equiv S$ and the functions q and ρ are simply the identity. We could alternatively attempt to generate such Poisson RVs by CDF inversion using a gamma latent process, but this method is far cheaper, as generating Poisson increments is computationally cheap (Devroye 1986).

The only immediately obvious alternative method uses IS (Section 2.5). Accordingly we devise an IS estimator. The question is how to choose the proposal distribution G_τ. We choose a distribution based on scaling the rate of each Poisson variate $X_1, \cdots, X_d$ by a common factor τ with $0 < \tau < 1$ that goes to zero as κ

goes to zero. Thus, under G_τ, each X_i is a Poisson RV with rate $\tau\lambda_i$,

$$G_\tau[X_i = k] = \frac{(\lambda_i\tau)^k \exp(-\lambda_i\tau)}{k!}, \quad k = 0, 1, 2, \ldots, d. \tag{3.39}$$

Thus, the IS estimator is given by

$$\hat{\ell}_{\mathrm{IS}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left\{\sum_{j=1}^{d} w_j X_j(\omega_i) \le \kappa\right\} \prod_{j=1}^{d} \frac{\exp\left(-\lambda_i(1-\tau)\right)}{\tau^{X_j(\omega_i)}}, \tag{3.40}$$

where for each replication, $\{X_j(\omega_i)\}_{j=1}^d$ are sampled independently according to (3.39). We choose the parameter τ such that the expected value of $\sum_{i=1}^d w_i X_i$ under (3.39) is equal to κ. It follows that τ is given by

$$\tau = \frac{\kappa}{\sum_{i=1}^d w_i \lambda_i}. \tag{3.41}$$

This IS estimator is, to the best of our knowledge, also novel.

We compare the results for the proposed quasi-monotone splitting method to that of the CMC and IS estimators in Table 3.7. The rates and the weights we set at $\lambda_i = 1 + (i-1) \times 0.2$ and $w_i = i$, $i = 1, \cdots, 12$. The number of samples per level is $\tilde{n} = 3000$. In order to estimate the variance of the quasi-monotone splitting estimator we run the algorithm $m = 200$ times and estimate the mean and the variance by sample mean and sample variance respectively. We use $m = 6 \cdot 10^6$ samples in the IS and the CMC algorithms. The CMC method is, as expected, unsatisfactory for $\ell \ll 1$. The quasi-monotone splitting and the IS estimators perform better. The WNRV values reveal that the IS estimator is slightly more efficient than the proposed quasi-monotone splitting estimator — beating it by a factor of two when $\kappa = 30$. In this case we have devised both the best (IS) and second-best (quasi-monotones splitting) estimators for this problem at once. The quasi-monotone splitting estimator is still more general, however. For example, if we need to calculate some quantity with respect to a more general quasi-monotone functional than a weighted sum, quasi-monotone splitting requires a trivial extension, but the IS method may be complex.

These problems are illustrative, but by no means exhaustive. Many other classes of practical problems have quasi-monotone forms. Network reliability prob-

Table 3.7: $\hat{\ell}$ for the sum of weighted Poisson RVs with $\lambda_i = 1 + (i-1) \times 0.2$ and $w_i = i$, $i = 1, 2, \cdots, 12$.

	CMC			Quasi-monotone splitting			Importance Sampling with $m = 6 \cdot 10^6$		
κ	$\hat{\ell}$	re($\hat{\ell}$)(%)	$\widehat{\text{WNRV}}(\hat{\ell})$	$\hat{\ell}$	re($\hat{\ell}$)(%)	$\widehat{\text{WNRV}}(\hat{\ell})$	$\hat{\ell}$	re($\hat{\ell}$)(%)	$\widehat{\text{WNRV}}(\hat{\ell})$
60	$1.02 \cdot 10^{-4}$	4.02	0.40	$1.07 \cdot 10^{-4}$	1.71	$1.62 \cdot 10^{-2}$	$1.06 \cdot 10^{-4}$	0.40	$3.20 \cdot 10^{-3}$
50	$2.00 \cdot 10^{-5}$	8.50	1.90	$2.26 \cdot 10^{-5}$	1.80	$1.82 \cdot 10^{-2}$	$2.33 \cdot 10^{-5}$	0.52	$5.20 \cdot 10^{-3}$
40	$4.83 \cdot 10^{-6}$	20.30	10.18	$3.97 \cdot 10^{-6}$	2.08	$2.69 \cdot 10^{-2}$	$3.97 \cdot 10^{-6}$	0.73	$1.00 \cdot 10^{-2}$
30	$8.33 \cdot 10^{-7}$	58.10	84.63	$4.80 \cdot 10^{-7}$	2.12	$3.75 \cdot 10^{-2}$	$5.07 \cdot 10^{-7}$	0.98	$1.82 \cdot 10^{-2}$

lems, (e.g. Gertsbakh and Shpungin 2016) and various network delay problems such as those induced by stochastic Project Evaluation and Review Techniques (e.g. Adlakha and Kulkarni 1989; Hagstrom 1990), i.e. PERT graphs, can be put into quasi-monotone form. Indeed, many network reliability problems are amenable to dynamic splitting, (Botev, L'Ecuyer, and Tuffin 2018; Botev et al. 2012) and these seem likely to be amenable to quasi-monotone splitting in particular. Various copula sampling problems (e.g. Embrechts, Lindskog, and McNeil 2003) can be reduced to quasi-monotone sampling problems, and indeed the generalization from Example 3.1 to restricted classes of Gaussian copulas, for example, is immediate. Many functionals are quasi-monotone with coordinate-wise positive vector arguments, e.g. L_p norms or more generally, weighted p-means, and Kolmogorov f-means and monotone functions of such means. We return to some particularly challenging problems based on these in Chapter 4. Quasi-monotone splitting can treat all estimation problems whose importance function arises from such functionals.

3.4 Asymptotics of quasi-monotone splitting

We take a moment to examine, through numerical simulation, the asymptotic behaviour of quasi-monotone splitting estimators. As mentioned in Section 2.3, in the rare event setting we are concerned with two types of asymptotic behaviour, both with regard to the rarity parameter as $\varepsilon \to 0$ (LRE) and to the effort parameter as $\eta \to \infty$ (ENRV/WNRV).

With regard to rarity asymptotics, we recall that the left quasi-monotone problems can employ $\varepsilon \equiv \kappa$ as a rarity parameter, and correspondingly, $\varepsilon \equiv 1/\kappa$ in the

right quasi-monotone case. We would like the estimator to attain LRE in the large effort limit for idealized problems, but have no analytic results for the more general case. Rearranging the definition of logarithmic efficiency (2.19) for an estimator $\hat{\theta}$ we see that ultimately, as $\varepsilon \to 0$,

$$\ln \mathbb{E}\left[\hat{\theta}_\varepsilon^2(\eta)\right] \approx 2 \ln \theta_\varepsilon(\eta). \tag{3.42}$$

We estimate this ratio by simulation. We use a quasi-monotone splitting estimator of ℓ with fixed total effort η and $t_k = k/20$ using 10% pilot effort and $\check{p} = 0.2$, estimating first and second moments empirically, across $R = 200$ replicates for each distinct value of κ. We can fit a simple linear model to the resulting $\log \hat{\mathbb{E}}[\ell^2] \approx \beta_0 + \beta_1 \log \hat{\ell}$, by ordinary least squares. Considering (3.42), the coefficient $\hat{\beta}_1$ serves as an estimator for the true quantity of interest, $\lim_{\ell \to 0} \frac{\mathrm{d}(\log \mathbb{E}[\ell^2])}{\mathrm{d}(\log \ell)}$. In the case that logarithmic efficiency is attained for an estimator and that the relationship between these variables is indeed linear and hence our estimates unbiased, we should find $\beta_1 \approx \lim_{\ell \to 0} \frac{\mathrm{d}(\log \mathbb{E}[\ell^2])}{\mathrm{d}(\log \ell)} = 2$. In practice, none of these criteria are fulfilled, and this approximation has unknown error. For comparison, we repeat the whole for two values of effort parameters $\eta = 10^5$ and $\eta = 10^3$. Holding effort fixed, we run the quasi-monotone splitting estimator $\hat{\ell}_\kappa$ while varying κ so that $\ell \to 0$. We repeat this procedure over 200 replications to estimate estimator distributions. In Figure 4.11 we plot these values for a variety of κ values for a representative partial-sum-of-order-statistics problem at two levels of effort. A line of best fit through the empirical estimates shows a gradient of $\hat{\beta}_1 = 1.997$ when $\eta = 10^5$, and $\hat{\beta}_1 = 1.902$ when $\eta = 10^3$, which is compatible with logarithmic efficiency in the rarity parameter.

Turning to large-effort asymptotics, we consider the effort-normalized relative variance, which we estimate empirically. For these simulations we fix all parameters apart from effort, and choose κ values for each problem such that the target event probabilities are comparable, with $\ell \approx 5 \cdot 10^{-15}$. We recall that under our assumptions, the ENRV and WNRV are eventually constant for a given model and set of parameters. We estimate WNRV over the replicates, calculating confidence intervals by bootstrap resampling. Numerical simulations plotted in Figure 3.6 show some signs of this statistic converging to constant WNRV although this is

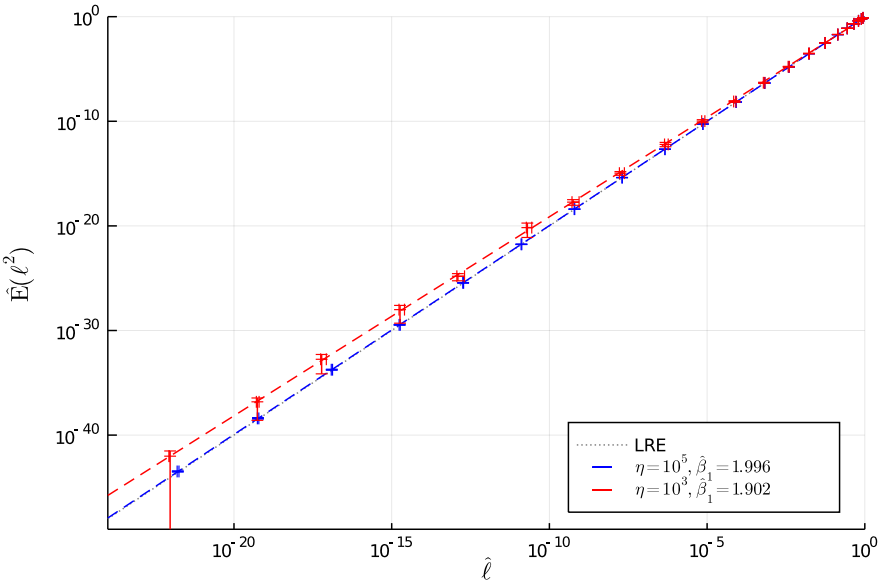


Figure 3.5: Small-probability-asymptotic behaviour in quasi-monotone splitting, for the partial sum of log-normal order statistics, with $D = 8$, $d = 4$, $\mu = 0$, $\sigma = 2$. over $R = 200$ replications. Error bars denote bootstrap 95% uncertainty intervals over 1000 replications.

not particularly fast or clear-cut, and the confidence interval for this statistic is wide.

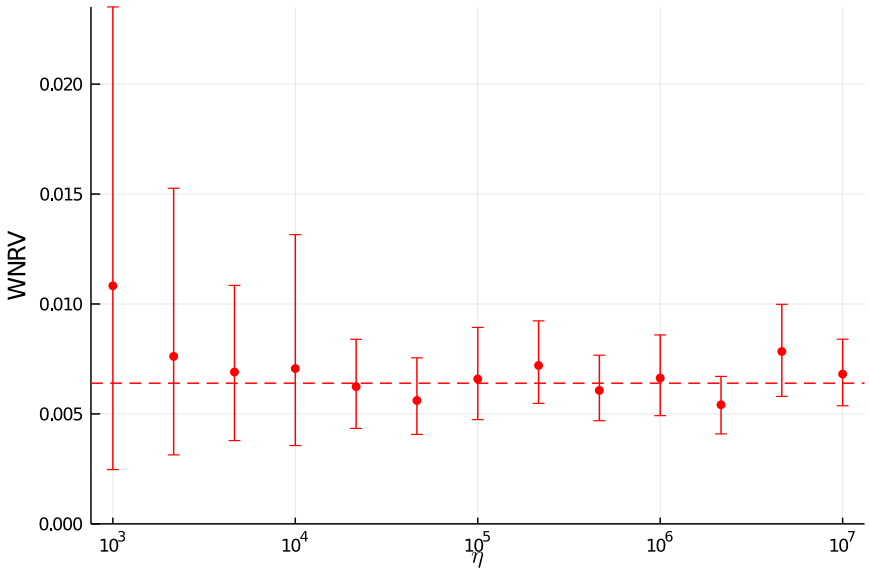


Figure 3.6: Large sample WNRV of quasi-monotone splitting, over $R = 100$ replications, for the partial sum of log-normal order statistics, with $D = 8$, $d = 4$, $\mu = 0$, $\sigma = 2$. Error bars shows 95% confidence interval over 1000 bootstrap replicates. Dotted line denotes inverse-variance-weighted sample mean.

3.5 A quasi-monotone rare-event conditional problem

We have asserted that a useful facility of quasi-monotone splitting is not just the tail-truncated probability estimation that all the examples in the chapter have used so far but also the estimation of rare-event conditional estimands. We demonstrate this by numerically estimating the (right-tailed) conditional excess (2.4) of the sum over independent variables $X_k, k = 1, \ldots, d$,

$$\theta(\kappa) = \mathbb{E}\left[\sum_{k=1}^{d} X_k \mid \sum_{k=1}^{d} X_k > \kappa\right]. \tag{3.43}$$

This is trivially a right-tailed quasi-monotone problem. It is a special case of, for example, the sum-of-random-variates model of Subsection 3.3.1. The target event here is $\mathcal{L}_\kappa \overset{\text{def}}{=} \{\sum_{k=1}^d \mathsf{X}_k > \kappa\}$ i.e., $S(\boldsymbol{x}) \overset{\text{def}}{=} \sum_{k=1}^d x_k$. We take the components to be independent log-normal variates. This problem in particular is of great interest in finance, for example in the pricing of arithmetic Asian options (e.g. Kemna and Vorst 1990). See also specialized approaches in, e.g. Botev, Salomone, and Mackinlay (2019).

A classic alternative method for this problem is the basic Gibbs sampler, which can also sample generically from this target set. A Gibbs sampler for such a right-tailed sum problem is straightforward. We can use the univariate conditional CDF (2.39),

$$F_{(k)}(x) = \mathbb{P}[\mathsf{X}_k > x \mid \mathsf{X}_1, \dots, \mathsf{X}_{k-1}, \mathsf{X}_{k+1}, \dots] \tag{3.44}$$

$$= \mathbb{P}[\mathsf{X}_k > x \mid \mathsf{X}_k > (\kappa - \sum_{j \neq k} \mathsf{X}_j)]. \tag{3.45}$$

That is, the conditional CDF $F_{(k)}$ is the distribution of $\mathsf{X}_k \sim F_k$ truncated to the range $((\kappa - \sum_{j\neq k} \mathsf{X}_j), \infty)$. For general random variates we may simulate such truncated random variates using quantile transforms. In this particular case, the log-normal distribution, we simulate variates as transforms of truncated normal variates. Simulating truncated normal variates is subject to various difficulties (Botev and L'Ecuyer 2017; Robert 1995). We use the rejection sampling method of Robert (1995) to simulate rare tail events with high accuracy.

When comparing the quasi-monotone splitting and random-sweep Gibbs sampling estimators for such a problem, we suspect the influence of the dimension d may be significant. Since the Gibbs sampling, unlike quasi-monotone splitting, updates dimensions separately, we might expect dimensionality to affect the methods differently. Accordingly we construct a series of problems of increasing dimension where for each dimension k the coordinate X_k is distributed

$$\mathsf{X}_k \sim \text{LogNormal}(1, k). \tag{3.46}$$

To keep the target events comparably rare across different dimensions we select

$$\kappa = 50 \sum_{k=1}^{d} \mathbb{E}[X_k] \tag{3.47}$$

$$= 50 \sum_{k=1}^{d} \exp(1 + k^2). \tag{3.48}$$

With this, we are able to perform numerical simulations of the relative performance of these estimators. Estimated corresponding $\hat{\theta}$ values are given in Table 3.8. Estimates are constructed via quasi-monotone splitting with $\eta_{\text{pilot}} = 10^5$, $\eta_{\text{main}} = 10^8$ with $R = 100$ replications. The Gibbs sampler is run until it generates $n = 10^7$ total samples. Initial values are chosen to be $x_k = (2\kappa + 1)/d, k = 1, \ldots, d$. Estimator variance in the Gibbs sampler is estimated by blocked means, dividing the chain into 100 blocks of size 10^5. [4] As the sampling methods here are not at all similar, we rely on WNRV to estimate an effort-parametrization-independent measure of efficiency.

Table 3.8: Estimator comparison for $\hat{\theta}$, the conditional excess, for the example problems.

		Gibbs			Splitting		
d	κ	$\hat{\theta}$	$\widehat{\text{re}}(\hat{\theta})$ (%)	$\widehat{\text{WNRV}}(\hat{\theta})$	$\hat{\theta}$	$\widehat{\text{re}}(\hat{\theta})$ (%)	$\widehat{\text{WNRV}}(\hat{\theta})$
1	224	282	0.0272	$2.66 \cdot 10^{-6}$	282	0.268	$1.42 \cdot 10^{-5}$
2	$1.23 \cdot 10^3$	$2.61 \cdot 10^3$	0.209	0.00016	$2.6 \cdot 10^3$	0.556	$6.62 \cdot 10^{-5}$
3	$1.35 \cdot 10^4$	$6.16 \cdot 10^4$	1.7	0.00903	$6.01 \cdot 10^4$	2.59	**0.00151**
4	$4.19 \cdot 10^5$	$4.62 \cdot 10^6$	6.85	0.149	$4.82 \cdot 10^6$	20.3	**0.0913**
5	$3.69 \cdot 10^7$	$1.13 \cdot 10^9$	9.31	**0.255**	$1.24 \cdot 10^9$	59.9	0.812

Our simulations here show the splitting method is comparable to the Gibbs sampler, outperforming Gibbs sampling for $1 < d \leq 4$ by this estimated WNRV metric. Unlike the Gibbs sampler estimator, the splitting estimator has simultaneously recovered an estimate $\hat{\ell} = \mathbb{P}[\mathcal{L}_\kappa]$, which is not in general possible for a Gibbs sampler.[5]

[4] As an implementation detail, he pilot run for these quasi-monotone simulations is generated using the non-adaptive method of Chapter 4, i.e., with fixed pilot splitting times $t'_k = k/10$.

[5] Although, one can also extract an estimate of ℓ from a Gibbs sampler under certain additional restrictions upon S; see Gudmundsson and Hult (2014).

3.6 Conclusion

In this chapter, we have proposed a dynamic quasi-monotone splitting estimator which can efficiently solve a broad class of time-independent problems of industrial interest. The method embeds a time-independent problem within a continuous-time Markov process so that the target distribution corresponds to the marginal distribution of the Markov process time $t = 1$. Estimates of functionals over this distribution may then be found with a simple dynamic splitting estimator with easy, and largely automatic, implementation. The resulting class of algorithms is broadly applicable. It is not necessarily competitively efficient with particular custom-designed Monte Carlo estimators for special-case problems. However, it can be applied to problems without known specific Monte Carlo estimators, requires only minimal and simple calculations, and is sometimes state-of-the-art in efficiency. Several problems of importance in wireless reliability engineering, for example, may be handled as quasi-monotone problems. We have left untouched certain questions about how close these methods are to optimality, and what a principled time selection method would look like. This question, we return to in the next chapter.

Chapter 4

Improving pilot run time selection

We return to the question of choosing splitting times for quasi-monotone problems, and develop improved estimators of those splitting times. We numerically investigate the importance of selecting time-steps well. We refine the method for time selection developed in Section 3.2 using the tools of extreme value theory in Section 4.1, and of survival analysis in Section 4.2. In Section 4.3 we perform numerical comparisons against the baseline and derive recommendations for choice of method.

We have asserted in previous chapters that splitting levels can affect the efficiency of quasi-monotone splitting. Here we refine the efficacy of our method for choosing the splitting levels by providing new alternatives, and quantifying the precision of each. To reprise, in Section 3.2 we developed a simple algorithm that attempts to enforce a constant conditional survival probability between levels with a value close to an target $\check{p}$, which, on the basis of certain idealization arguments, should lead to the smallest variance estimator. This is equivalent to selecting splitting times $t^{(m)}, m = 1, \ldots, M$ so that $\mathbb{P}[S_g(\mathsf{G}(t_m)) \leq \kappa \mid S_g(\mathsf{G}(t_{m-1})) \leq \kappa] = \check{p}$. We can attain this using the lifetime distribution $\mathsf{T} \sim T$ of the random variate $\mathsf{T} = \inf\{t : S_g(\mathsf{G}(t)) > \kappa\}$. More precisely, we need to estimate the inverse CCDF $\bar{T}^{-1}$. T optimal times are given by (3.29) as $t_m = \bar{T}^{-1}(\check{p}^m)$. With this in mind we attempt to estimate the survival distribution T to find these times.

Throughout this chapter we use a set of contrived problems as test cases for the efficacy of our method. These problems provide examples of different estimator

behaviour, in particular with regard to optimal splitting times. Each is a left quasi-monotone problem with the usual target set, $\mathcal{L}_\kappa \stackrel{\text{def}}{=} \{S(\boldsymbol{X}) \leq \kappa\}$.

Example 4.1 (LNSum(κ))**.** LNSum is a left quasi-monotone estimation problem concerning a vector of independent log-normal random variables, where $\boldsymbol{X}_i \sim \text{LogNormal}(1, i)$. The importance function is given by a simple sum over the components, $S(\boldsymbol{X}) = \sum_{i=1}^{5} \boldsymbol{X}_i$.

Example 4.2 (PPNorm(κ))**.** PPNorm is a left quasi-monotone problem concerning a vector whose components are independently distributed as generalized Pareto random variates, $\boldsymbol{X}_i \sim \text{GPD}((7+i)/2, 2), i = 1, \ldots, 7$. The importance function is the p-norm of the vector $S(\boldsymbol{X}) = \|\boldsymbol{X}_i\|_{100}$.

Example 4.3 (PPMetric(κ))**.** Here we use the same random vector as PPNorm but in PPMetric(κ), the importance function now uses a different p-norm (or more properly, metric, as it is no longer a true norm) with $p = 1/10$. It is otherwise identical, i.e. $S(\boldsymbol{X}) = \|\boldsymbol{X}_i\|_{1/10}$.

Where we do not assume a particular value for κ, we suppress it and simply refer to, for example, PPNorm. All three example problems have intractable tail distributions, in the sense of having no apparent analytic form for tail probabilities $\hat{\ell} = \mathbb{P}[\mathcal{L}_\kappa]$. They all have a simple quasi-monotone form. We take, for each of these, $\{\boldsymbol{G}(t)\}_t \sim \text{GammaProc}(1,1)^{\times d}$, and $S_g \stackrel{\text{def}}{=} S \circ \boldsymbol{q}_{\text{inc}}$.

We use a small set of comparable κ values for each of these problems. Estimated corresponding $\hat{\ell}$ values are given in Table 4.1.

By way of illustration we plot the CCDF $\bar{T}$ of the lifetime distributions of the quasi-monotone splitting samples for each of these problems in Figure 4.1. These curves are estimated using the method of Section 4.2, to be introduced momentarily, with a large effort parameter so as to reduce error. Note that between problems and also between κ values, the associated CCDFS have different shapes and thus different ideal splitting times in our quasi-monotone splitting design.

In this chapter, as in Section 3.2, we use a pilot run to estimate $\bar{T}$. Here we modify the structure of the pilot run; rather than the *ad hoc* adaptive method of Algorithm 3.2 we use a pilot run which is identical to the usual fixed effort quasi-monotone splitting method of Algorithm 3.1. We choose equally-spaced initial

Table 4.1: $\hat{\ell}$ for the example problems, $t'_k = k/64$, $\tilde{n} = 2^{23}$. Relative error estimate uses sample variance from $R = 16$ replications.

model	κ	$\hat{\ell}$	$\hat{\mathrm{re}}(\hat{\ell})$
LNSum	0.02	$4.73 \cdot 10^{-15}$	0.0852
PPNorm	0.1	$7.96 \cdot 10^{-15}$	0.0669
PPMetric	$2 \cdot 10^{7}$	$5.33 \cdot 10^{-15}$	0.354
LNSum	0.2	$1.4 \cdot 10^{-7}$	0.0332
PPNorm	1	$1.52 \cdot 10^{-7}$	0.0431
PPMetric	$2 \cdot 10^{8}$	$7.09 \cdot 10^{-8}$	0.168
LNSum	2	0.00317	0.0258
PPNorm	10	0.00614	0.0223
PPMetric	$2 \cdot 10^{9}$	0.00871	0.0291

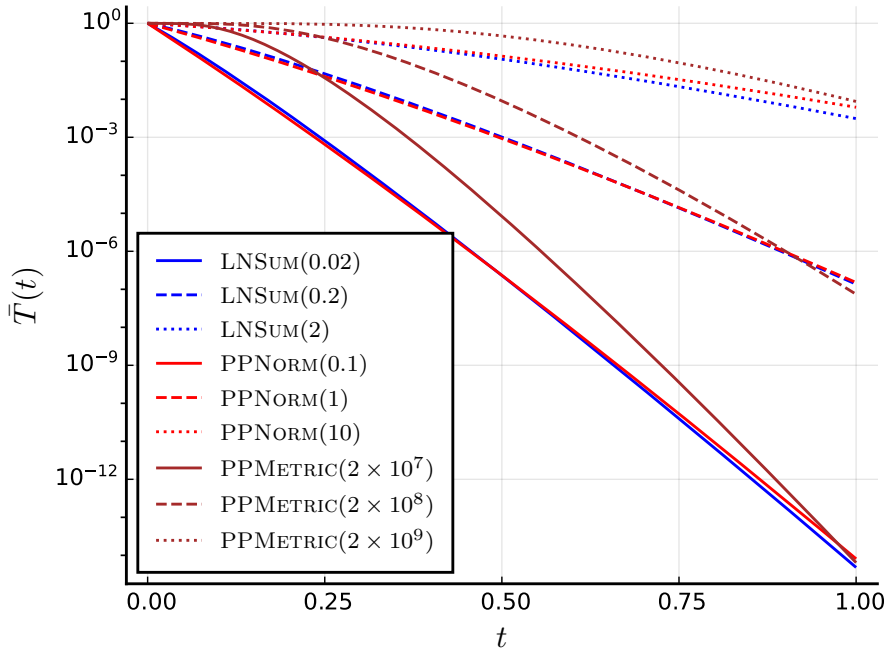


Figure 4.1: Estimated CCDFs for example problems, calculated by the survival method with $\tilde{n} = 2^{23}$, $t'_k = k/64$.

times $t'_k = k/K, i = 1 \ldots, K$ and per-level effort $\tilde{n}'$. The virtue of this approach is that we have a more transparent effort parameterization $\eta_{\text{pilot}} = \tilde{n}'K$ which explicitly upper bounds the number of random simulations. In the *ad hoc* adaptive method the run time arises from an opaque combination of search parameters and the problem. Performing like-for-like comparisons across estimators in a predictable manner is difficult in such a setting. We prefer to side-step this problem by altering the setup. In the main run we assign a per-level effort budget of $\tilde{n}$ at M times, and likewise an associated effort of $\eta_{\text{main}} = \tilde{n}M$. Where we have a compound splitting estimator comprising a pilot run with per-level effort $\tilde{n}'$ and K levels, which contributes $\eta_{\text{pilot}} = \tilde{n}'K$ realizations, the total effort is upper bounded by $\eta = \eta_{\text{main}} + \eta_{\text{pilot}} = \tilde{n}M + \tilde{n}'K$. Assuming that the expected cost of simulating the latent processes remains close to constant on average across different parameter ranges, which we observe in practice, this gives us a reasonable effort parameter with which to compare different quasi-monotone splitting estimators. Accordingly we favour effort-normalized measures like ENRV over WNRV to compare estimator parameters within a given model.

We have discussed η_{pilot} and η_{main} as effort parameters which suggests that the execution time is deterministic. This is not strictly true, because it ignores the possibility of early extinction. Strictly, these effort parameters upper bound the total number of, respectively, pilot run and main run random variate realizations. The splitting method may terminate early if the population undergoes an extinction, i.e., when no particles remain in $\mathcal{L}^{(m)}$ at step $m < M$. If this occurs in either the pilot or main run, the total number of random variate realizations is smaller than η. Since we expect premature extinctions to be rare in a splitting algorithm specifically design to make the conditional survival probability large, we use the approximation that the variation in effort due to premature extinctions does not affect the linear computational scaling in this parameter. Where effort is low, however, this approximation may be poor. In the pilot run in particular, the number of particles is low and the chance of premature extinction may be high. There is an additional complication with extinction in the pilot run, which is that since $\hat{\ell}^{(m)} = 0$ for some $m < M$, they lead to a non-invertible estimate $\hat{\tilde{T}}$ and hence undefined splitting times. To address this possibility we give, for each of the splitting time estimators, a reasonable smoothing which returns an invertible

$\hat{\bar{T}}$ even in the case of extinction. This choice of a non-adaptive pilot run in combination with small pilot effort increases the overall error — where the non-adaptive pilot run suffers premature extinction, the estimated splitting times are far from optimal. By contrast, in the adaptive method such an outcome is disallowed, and extinctions lead instead to a random increase in pilot effort as new pilot times are added. Our choice here to set aside adaptive pilot times, has exaggerated performance differences between methods, although it should not change the ranking of the methods. All methods here can be directly converted to an adaptive version in a situation where we do not need to maintain such a strict effort parameterization for the sake of comparison.

In the previous chapter we constructed estimates $\bar{T}(t_k) = \ell^{(k)}$ as a side-effect in Algorithm 3.1 (Line 16) using initial counts at the start of each time interval, $\tilde{n}'^{(0)}, \ldots, \tilde{n}'^{(K-1)}$ and pruned survival counts at the end of each interval, $n'^{(1)}, \ldots, n'^{(K)}$. In this sequel we once again use the same basic setup, changing the methods to construct the estimated $\hat{\bar{T}}$. We regard the recorded counts as realizations of random variables, $n'^{(k)} \sim \mathsf{N}'^{(k)}$ which count survival of random paths of the splitting method, and fit statistical models to these. We present two alternative estimation methods, based upon extreme value theory and survival analysis. These two alternatives constitute the chief point of difference between the previous approach and our earlier approach. As before, all the estimators of $\hat{\bar{T}}$ employ the approximation used in our idealization arguments, that the data from our pilot simulation represents independent observations drawn from the target distribution. Specifically, we assume that each of our particle lifetimes are drawn from $\mathsf{T} \sim T$ and that the observations $\boldsymbol{\xi}(t_m) \in \mathcal{G}^{(m)}, m = 1, 2, \ldots, M'$ are mutually independent, which is not in general true.

A point of difference between this chapter and the previous is that we modify the spacing of estimating splitting times in the main run. In the prior version (3.30) we chose

$$M = \lceil \log \hat{\bar{\ell}} / \log \check{p} \rceil. \tag{4.1}$$

then back-substituted into (3.29) to find the target splitting times. However, this ignores that for a given target probability $\ell = [\mathcal{L}]$ only certain values of $\check{p}$ are feasible as a step size — specifically, those such that $\check{p}^M = \ell$ for some $M \in \mathbb{N}$.

Previously we solved this problem by tolerating a different conditional probability for the final step of $p^{(M)} < \check{p}$. An alternative method adjusts $\check{p}$ to a feasible value, given the preliminary estimate of $\hat{\ell}$ and the corresponding M, by setting

$$\check{p}^* = \sqrt[M]{\hat{\ell}'}. \tag{4.2}$$

We refer to this as *squeezing* $\check{p}$. In the case that we have a noiseless estimate of the CCDF this perturbs us away from the target $\check{p}$ splitting probability but leads to more uniform splitting probabilities. Empirically, this change makes negligible difference to estimator variance in any of the problems considered here, but we will need to be precise about this matter when considering the estimation of conditional survival probabilities, later. Throughout this chapter we squeeze splitting estimators except where otherwise stated.

4.1 Extreme value method

Extreme Value Theory (EVT) studies the right tails of univariate distributions, furnishing us with limiting distributions of the tail of various sequences of random variables (McNeil, Frey, and Embrechts 2005) in particular those arising from heavy-tailed random variable. EVT results tell us that many distributions are "similar" in the sense that their right tail distribution over sum threshold eventually approaches a member of the Generalized Pareto Distribution (GPD) family as the threshold increases.This motivates a simple parametric model for optimal splitting times in which we assume that its dynamics are well modelled by some GPD. In the extreme rarity regime where the estimation task is at its most challenging, this provides a low-complexity parametric approximation for the CCDF for the lifetime distribution T which we estimate to understand tail behaviour (McNeil 1997). We introduce background to this method here.

Definition 4.1 (Excess distribution over threshold u)**.** Let T be an rv with law T. The excess distribution over the threshold u has law

$$T_u(t) = \mathbb{P}[\mathsf{T} - u \leq t \mid \mathsf{T} > u] = \frac{T(t+u) - T(u)}{1 - T(u)} \tag{4.3}$$

for $0 \leq t < t_T - u$, where $t_T \leq \infty$ is the right endpoint of T.

Parenthetically, we note that this is equal to the definition of the conditional excess over threshold in (2.4) up to a translation by u, although that parallel is not exploited here.

Definition 4.2 (Generalized Pareto distribution)**.** For $\nu, \beta > 0, \mu \in \mathbb{R}$ the CDF of a Generalized Pareto distribution (GPD) is given by

$$G_{\nu,\beta,\mu}(t) = \begin{cases} 1 - (1 + \nu(t-\mu)/\beta)^{-1/\nu} & \text{if } \nu \neq 0 \\ 1 - \exp(-(t-\mu)/\beta) & \text{if } \nu = 0. \end{cases} \tag{4.4}$$

The support of this distribution is

$$\begin{aligned} T \sim G_{\nu,\beta,\mu} \Rightarrow & \\ \operatorname{supp}(T) = (\{t \geq \mu\} \cap \{\nu \geq 0\}) & \qquad (4.5) \\ \cup \left(\{\mu \leq t \leq \mu - \beta/\nu\} \cap \{\nu < 0\}\right). & \qquad (4.6) \end{aligned}$$

In numerical estimation the case ν arises with probability 0 (and in any case may be recovered as a limit as $\nu \to 0$) so we hereafter suppress this possibility (4.4), abbreviating

$$G_{\nu,\beta,\mu}(t) = 1 - (1 + \nu(t-\mu)/\beta)^{-1/\nu}. \tag{4.7}$$

The main result of use to our ends from EVT is the Pickands-Balkema-de Haan theorem (Balkema and de Haan 1974; Pickands III 1975).

Theorem 4.1 (Pickands-Balkema-de Haan)**.** We can find a function $\beta(u)$ such that

$$\lim_{u \to t_T} \sup_{0 \leq t < t_T - u} \left| T_u(t) - G_{\nu,\beta(u),0}(t) \right| = 0$$

if (and only if) T is in the *maximal domain of attraction* of the extreme value distribution with parameter ν for some $\nu \in \mathbb{R}$.

This maximal domain of attraction was introduced in the Fisher-Tippett theorem (Fisher and Tippett 1928), and is well-analysed in the EVT literature (e.g. Embrechts, Kluppelberg, and Mikosch 1997). For the current purposes it is suffi-

cient to note that it contains most distributions in use in statistics. We assume in particular that all T arising in our quasi-monotone splitting problems are in this domain.

This motivates the modelling of the tail with a GPD. For a first attempt at this method we will in fact model not just the tail but the entire survival time distribution by a GPD, $T \sim G_{\nu,\beta,\mu}(t) \stackrel{\text{def}}{=} 1 - \left(1 + \frac{\nu(t-\mu)}{\beta}\right)^{-1/\nu}$. Then for $t > s \geq 0$ and assuming that $G_{\nu,\beta,\mu}(s) > 0$, the survival probability over an interval $(s, t]$ is

$$\begin{aligned}
p_{s,t,\nu,\beta,\mu} &\stackrel{\text{def}}{=} \mathbb{P}[T \geq t \mid T > s] && (4.8)\\
&= \frac{\mathbb{P}[T \geq t \cap T > s]}{\mathbb{P}[T > s]} && (4.9)\\
&= \frac{\mathbb{P}[T \geq t]}{\mathbb{P}[T > s]} && (4.10)\\
&= \frac{\bar{G}_{\nu,\beta,\mu}(t)}{\bar{G}_{\nu,\beta,\mu}(s)} && (4.11)\\
&= \frac{\left(1 + \frac{\nu(t-\mu)}{\beta}\right)^{-1/\nu}}{\left(1 + \frac{\nu(s-\mu)}{\beta}\right)^{-1/\nu}} && (4.12)\\
&= \left(\frac{\beta + \nu(s-\mu)}{\beta + \nu(t-\mu)}\right)^{1/\nu}. && (4.13)
\end{aligned}$$

Under this assumption, the observations are once again binomially-distributed survival counts over the increments of our pilot run as in Subsection 2.6.4, but now the binomial probability parameter has a parametric form arising from the GPD. Explicitly, in the kth step, the likelihood of each observation interval $(t'_k, t'_{k+1}]$ is given by the binomial probability mass function for $N'^{(k)}$ survivals from a trial of size $\tilde{N}'^{(k-1)}$. Thus the likelihood of any given interval is

$$\begin{aligned}
L(m; \nu, \beta, \mu) &= \mathbb{P}[N'^{(k)} = n'^{(k)} \mid \tilde{N}'^{(k)} = \tilde{n}'^{(k-1)}; \nu, \beta, \mu] && (4.14)\\
&= \binom{\tilde{n}'^{(k-1)}}{n'^{(k)}} p_{t'_{k-1},t'_k,\nu,\beta,\mu}^{n'^{(k)}} \bar{p}_{t'_{k-1},t'_k,\nu,\beta,\mu}^{\tilde{n}'^{(k-1)} - n'^{(k)}} && (4.15)\\
&= \binom{\tilde{n}'^{(k-1)}}{n'^{(k)}} p_k^{n'^{(k)}} \bar{p}_k^{\bar{n}'^{(k)}}. && (4.16)
\end{aligned}$$

For compactness we have introduced $\bar{n}'^{(k)} \stackrel{\text{def}}{=} \tilde{n}'^{(k-1)} - n'^{(k)}$ and $p_k \stackrel{\text{def}}{=} p^{n'^{(k)}}_{t'_{k-1},t'_k,\nu,\beta,\mu}$. Under the usual idealization assumptions the observations are independent so the the joint likelihood is approximated

$$L(\mathcal{D};\nu,\beta,\mu) = \mathbb{P}[\tilde{n}'^{(0)},\ldots,\tilde{n}'^{(K-1)},n'^{(1)},\ldots,n'^{(K)};\nu,\beta,\mu] \tag{4.17}$$

$$\approx \prod_{k=1}^{K} \mathbb{P}[\mathsf{N}'^{(k)} = n'^{(k)} \mid \tilde{\mathsf{N}}'^{(k-1)} = \tilde{n}'^{(k-1)};\nu,\beta,\mu]. \tag{4.18}$$

$\mathcal{D}$ here denotes the observed simulation output, $\mathcal{D} \stackrel{\text{def}}{=} \{\tilde{\mathsf{N}}^{(k-1)},\mathsf{N}^{(k)},k=1,\ldots,K\}$. The method of maximum likelihood estimates the parameters as

$$\widehat{(\nu,\beta,\mu)} = \operatorname*{argmax}_{\nu,\beta,\mu} \log L(\mathcal{D};\nu,\beta,\mu). \tag{4.19}$$

where

$$\log L(\mathcal{D};\nu,\beta,\mu) = \sum_{k=1}^{K}\left(\log\binom{\tilde{n}'^{(k-1)}}{n'^{(k)}} + n^{(k)}\log p_k + \bar{n}^{(k)}\log\bar{p}_k\right). \tag{4.20}$$

This formula has no apparent explicit form for the maximiser. We solve it numerically using automatic second-order gradient-based optimization (Mogensen and Riseth 2018; Mogensen et al. 2020) with forward-mode automatic differentiation (Griewank and Walther 2008; Rall 1981; Revels, Lubin, and Papamarkou 2016) to find gradients efficiently.

This model entails strong assumptions which we do not suppose our models meet in general. A GPD fit is in typical applications only considered for excess distribution (Definition 4.1), $\mathsf{T}-u \mid \mathsf{T} \geq u$, and a more complete fitting procedure would in addition estimate the threshold u which gives us the value beyond which the asymptotic approximation to a GPD-distribution is close enough to be useful. For values of $\mathsf{T} < u$, the "body" of the distribution, we have no reason to expect good performance fitting to the GPD model. In lifetime estimation problems we would typically prefer another model for the body of the life distribution, constructing the final estimator as a mixture of the extreme-value tail model and that other body model. The other estimator introduced in this chapter, Section 4.2, for example, would fit this purpose. For the instrumental goal of selecting splitting

times from a pilot run well enough for the main run, we avoid such elaborate procedures. Various case studies in methods of that kind are available in Embrechts, Kluppelberg, and Mikosch (1997), Markovitch and Krieger (2002), and McNeil (1997) and references therein.

As a heuristic means of fitting GPD robustly despite possible ill fit in some parts of the survival curve, we use a simple, weighted maximum likelihood procedure. Weighted maximum likelihood estimates (Hu and Zidek 2002; Wang 2001) minimise the influence of a bad fit for some observations by assigning a low weight to those. In our case, this means updating (4.20) with weight function $w : \{1, 2, \ldots, K\} \to [0, \infty)$,

$$\widehat{(\nu, \beta, \mu)}_w \stackrel{\text{def}}{=} \sum_{k=1}^{K} w(m) \log L(m; \nu, \beta, \mu). \tag{4.21}$$

We use *ad hoc* weights, simply setting $w(m) \stackrel{\text{def}}{=} t'_m$ so that the tail values are more important in the overall fit. More complicated schemes are possible, e.g., iteratively reweighting likelihoods; see, for example, Green (1984), Holland and Welsch (1977), and Street, Carroll, and Ruppert (1988).

GPD models have well-known estimation challenges. Different methods are required to attain convergence for different regions of parameter space and the maximum likelihood estimator may even fail to exist for some parameter values (Grimshaw 1993; Hosking and Wallis 1987; Hüsler, Li, and Raschke 2011; McNeil 1997). Standard regularity requirements for maximum likelihood estimation are not satisfied: the support of this distribution depends upon its parameters (4.6). We impose restrictions upon parameter ranges which obviate this problem.

Since we know by construction the support of T includes 0, allowing $\mu \neq 0$ is suspect. We fix $\mu \equiv 0$. We could lift this restriction if we used a mixture model, as mentioned earlier, wherein the GPD was fit only to the tail. We also exclude negative ν values. $\nu < 0$ implies that T has light tails, and in particular, has bounded support, which we know is false in the quasi-monotone problem for all latent processes without positive drift for which the support is unbounded (3.26). We thus set $\nu > 0$. Over this parameter range, the maximum likelihood estimates are known to be asymptotically normal and asymptotically efficient (Grimshaw

1993; Smith 1985). Under these assumptions, (4.13) simplifies to

$$p_{s,t,\nu,\beta} = \left(\frac{\beta + \nu t}{\beta + \nu s}\right)^{-1/\nu}. \tag{4.22}$$

In pilot runs, the population of particles may go extinct with non-zero probability. In such cases, the maximum likelihood estimate is necessarily $\nu < 0$, which we have excluded *a priori*. We solve this problem by an *ad hoc* procedure, regularising the likelihood using Laplace smoothing of the binomial survival counts. That is, we replace our actual observations $\mathcal{D}$ with a Laplace smoothed version, $\mathcal{D}_{\text{smoothed}}$. In $\mathcal{D}_{\text{smoothed}}$ we update the survival counts as $\tilde{n}'^{(k)}_{\text{smoothed}} = \tilde{n}'^{(k)} + 2$ and $n'^{(k)}_{\text{smoothed}} = n'^{(k)} + 1$. This method leads to well-defined, if additionally biased, splitting time estimates, even where the splitting method suffers premature extinction.

Plugging estimated parameters based on smoothed data and reweighted likelihoods in into (4.7) we estimate the lifetime distribution as $\hat{T} = \hat{G} \overset{\text{def}}{=} G_{\widehat{\nu,\beta,0}}$. We read off the desired splitting times as per (3.29). This yields

$$\log \hat{\bar{G}}(t_m) = m \log \check{p} \tag{4.23}$$

$$\log(1 + \hat{\nu}(t_m)/\hat{\beta}) = -\hat{\nu} m \log \check{p} \tag{4.24}$$

$$1 + \hat{\nu}(t_m)/\hat{\beta} = \exp(-\hat{\nu} m \log \check{p}) \tag{4.25}$$

$$t_m = \frac{\hat{\beta}}{\hat{\nu}}(\exp(-\hat{\nu} m \log \check{p}) - 1) \tag{4.26}$$

for $m = 1, 2, \ldots, M$. Comparative results using this method are presented in Section 4.3.

4.2 Splitting times via survival analysis

We revisit the method of Section 3.2, where we used linear interpolation to construct our $\bar{T}$ estimate. We introduce here the tools of survival analysis via hazard functions which provide a theoretical motivation for a superior estimator of splitting times. These motivate using linear interpolation over the log-domain CCDF, in order to improve the performance of the splitting time estimator with respect

to the linear CCDF estimate.

Survival function theory provides a methodology for estimating the *survival function* $\bar{T}(t) \stackrel{\text{def}}{=} \mathbb{P}[\mathsf{T} > t]$ of a non-negative scalar random variable $\mathsf{T} \sim T$. This survival function we can recognise as precisely the CCDF of the lifetime distribution, $\bar{T}$. As the quasi-monotone splitting method is by construction a method of estimating densities in terms of the lifetimes of simulated particles, this approach arises naturally.

We define some useful quantities.

Definition 4.3 (Hazard function)**.** The *hazard function* is given

$$h(t) \stackrel{\text{def}}{=} \frac{\mathrm{d}T(t)}{\mathrm{d}t}\frac{1}{\bar{T}(t)}. \tag{4.27}$$

This function is by inspection, non-negative, and gives the infinitesimal probability of a death at time t conditional upon it not having occurred so far. A death in this context for a latent particle is defined as that particle leaving the target event $\mathcal{L}$. The hazard function, aside from the constraint that it be non-negative, may be an arbitrary function which makes it a natural candidate for nonparametric estimation. Given a population of particles started at time 0 with lifespans distributed as T, we think of h as the conditional intensity of death at time t given that it has not yet occurred.

Definition 4.4 (Cumulative hazard function)**.** The *cumulative hazard function* H is given

$$H(t) \stackrel{\text{def}}{=} \int_0^t h(s)\mathrm{d}s. \tag{4.28}$$

Over intervals of time $[t, u]$ we use the cumulative hazard *increment*

$$H(t,u) \stackrel{\text{def}}{=} \int_t^u h(s)\mathrm{d}s = H(u) - H(t) \tag{4.29}$$

and the survival ratio

$$\bar{T}(t,u) \stackrel{\text{def}}{=} \frac{\bar{T}(u)}{\bar{T}(t)}. \tag{4.30}$$

We use some additional relations which follow as immediate consequences of these:

$$\bar{T}(t) = \exp[-H(t)] = \frac{j(t)}{h(t)}. \tag{4.31}$$

$$\bar{T}(t,u) = \frac{\exp[-H(u)]}{\exp[-H(t)]} = \exp[-H(t,u)]. \tag{4.32}$$

In terms of the cumulative hazard function, the desired splitting times of equation (3.29) become, by the substitution of (4.32),

$$-H(t_m) = m \log \check{p} \tag{4.33}$$

$$\Rightarrow t_m = \hat{H}^{-1}(-m \log(\check{p})). \tag{4.34}$$

Up to a sign change and a logarithm this is the CCDF. This small change, the switching to log-CCDF estimation, turns out to be important, as the estimation theory for the hazard function is better-behaved than that for linear CCDF interpolation. These two representations of the CCDF are depicted depicted in Figure 4.2b.

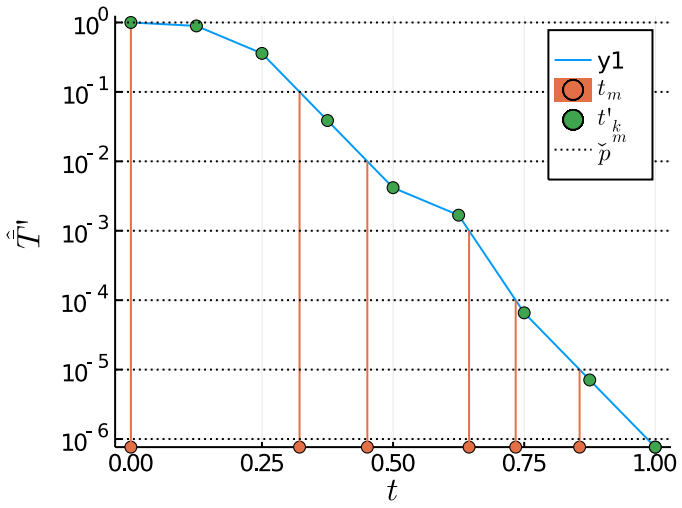
The major result of survival theory of use to us is unbiased estimators for some of these quantities. In particular, we may estimate cumulative hazard increments via the *life table* method (e.g. Aalen, Borgan, and Gjessing 2008). Suppose we have a population of particles, and the life span of the ith member of the population is given independently as $\mathsf{T}_i \sim T$. Let the random process $\{\mathsf{N}(t)\}_t$ count the number of surviving particles at each instant $\mathsf{N}(t) = \sum_i \mathsf{T}_i > t$. The life table estimate of a survival ratio is unbiased and is given

$$\hat{\bar{T}}(t_0, t_1) = \frac{\mathsf{N}(t_1)}{\mathsf{N}(t_0)}. \tag{4.35}$$

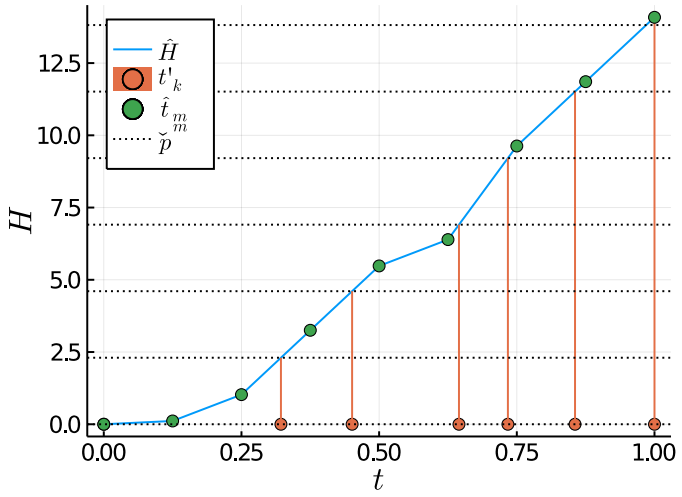
We use this to estimate

$$\hat{\bar{T}}(t_{k-a}, t_k) = \frac{n(t_k)}{\tilde{n}(t_{k-1})}. \tag{4.36}$$

Unbiased estimates of the survival function increments do not translate automatically to good estimates of the quantiles. The relationship between CCDF estimators and the quantile estimators that optimal splitting times is complicated



(a) In CCDF representation



(b) In hazard function representation

Figure 4.2: Stylized depiction of method of time selection using hazard function interpolation, using the same data as Figure 3.4.

and depends upon the family of distributions in question (Makarov 2006). On the other hand, the linear CCDF method we used in Section 3.2 has even weaker guarantees, so we hope for some empirical improvement.

With these survival analysis tools in hand, we return to the problem of splitting time estimation. A simple estimate of splitting times arises from the hazard function and thence the CCDF. This uses the hazard increment estimates from the pilot simulation to construct a hazard interpolation. Suppose we observe population survival statistics over the time interval $(t'_k, t'_{k+1}]$, as in a single step of a splitting simulation. Plugging (4.35) in to (4.34), we obtain cumulative hazard increment estimates over $k = 0, 1, \ldots, K-1$ for pairs

$$\hat{H}(t'_k, t'_{k+1}) = -\log \hat{\bar{T}}(t'_k, t'_{k+1}) \tag{4.37}$$

$$= \log \frac{\tilde{n}'(t'_k)}{n'(t'_{k+1})}. \tag{4.38}$$

The introduction of additional assumptions allows us to estimate an entire cumulative hazard function. We take $h : \mathbb{R} \to \mathbb{R}$ to be piecewise constant

$$\hat{h}(t) = \sum_k \mathbb{I}\{t'_{k-1} < t \leq t'_k\} \hat{h}_k. \tag{4.39}$$

for

$$\hat{h}_k = \frac{\hat{H}(t'_{k-1}, t'_k)}{t'_k - t'_{k-1}}. \tag{4.40}$$

From this we construct estimates $\hat{H}$ of the whole function H by integration, as per equation (4.28),

$$\hat{H}(t) = \int_0^t \hat{h}(s) \mathrm{d}s. \tag{4.41}$$

The resulting estimator is a linear interpolant with knots at times $0, t'_1, t'_2, \ldots, t'_K$. Plugging $\hat{H}$ in to (4.34) we obtain a survival function estimate,

$$\hat{\bar{T}}(t) = \exp(-\hat{H}(t)). \tag{4.42}$$

As in Section 4.1 we choose splitting level estimates to satisfy (3.30) and (3.29),

which in terms of a hazard function gives

$$\hat{t}_m = \hat{H}^{-1}(-\log(\check{p}^m)),\ m = 1, \dots, M. \tag{4.43}$$

Solving for t_m in (4.34) requires us to invert $\hat{H}$. Note that if $0 < n(t'_k) < \tilde{n}(t'_{k-1})$ for all m, then $\hat{H}$ is strictly continuous, strictly increasing and hence invertible. Further, the inverse of a piecewise linear function is still piecewise linear, and we can directly calculate $\hat{H}^{-1}$ by linear interpolation once again, with knots $\hat{H}(0) = 0, \hat{H}(t_1), \hat{H}(t_2) \dots, \hat{H}(t_M)$ and corresponding values $0, t_1, t_2, \dots, t_M$.

As in Section 4.1 we face a problem with regularity of the pilot run estimate; if $n'^{(k)} = 0$ for any k then the hazard function is non-invertible and the splitting times become ill-defined. Once again we avoid this circumstance using Laplace regularization of the Binomial counts, modifying the counts to be inflated pseudo-counts $\tilde{n}'^{(k)}_{\text{smoothed}} = \tilde{n}'^{(k)} + 2$ and $n'^{(k)}_{\text{smoothed}} = n'^{(k)} + 1$ as before. With this change we are able to provide meaningful estimates of the splitting times even for pilot runs suffering from extinction by linear extrapolation from the existing data.

4.3 Estimators in numerical comparison

We have various free parameters in in our splitting algorithm, including effort allocation, choice of splitting method and target survival probability. In this section we compare the effects of each of these by numerical study.

4.3.1 Effect of target survival probability $\check{p}$ on accuracy

Recalling that the optimal choice of target survival probability $\check{p}$ in Subsection 2.6.4 hinges upon various approximations and assumptions that do not hold in practice, we examine the sensitivity of the method to this parameter numerically. For each set of model parameters we share a single pilot estimate across all simulations. We use a large pilot effort budget $\eta_{\text{pilot}} = 32 \cdot 10^5$ with pilot times $t'_m = m/32$, and use the survival method Figure 4.1 to estimate $\hat{\hat{T}}$ with high precision. We use the same high precision $\hat{\hat{T}}$ estimate for given model parameters across all replications. Choosing a fixed target $\check{p}$, we estimate $t_1, \dots, t_M$ from the high precision

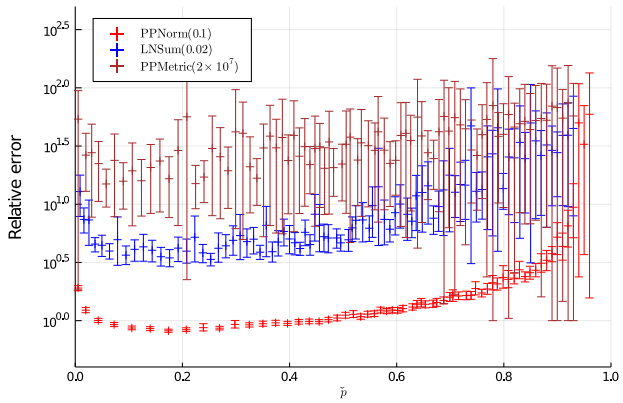
$\hat{\bar{T}}$. The resulting estimates should be close to $p^{(m)} \approx \check{p}$, $m = 1, \ldots, M$. In the main simulation we use a smaller simulation budget of $\eta_{\text{main}} = 10^3$. The resulting estimator should have low error in splitting times, and thus we hope variance in overall accuracy will be dominated by variance arising from the effect of our choice of $\check{p}$.

We do observe diverse responses to this parameter in the example problems. Results for all feasible values $0.01 \leq \check{p} \leq 0.97$ that differ by at least 0.005 are shown in Figure 4.3. We observe that the relatively good values lie in the range $\check{p} \in [0.1, 0.3]$ for all problems. It is not clear that there exists a consistent optimal value of $\check{p}$ across models, or across κ values within a model, although values around $\check{p} = 0.2$ do reasonably well in all problems. This numerical result motivates a choice of $\check{p} = 0.2$ throughout this chapter.

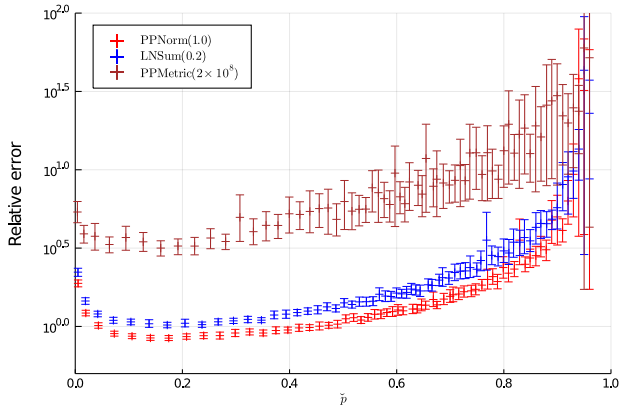
4.3.2 Effect of time selection method

Here we compare the two new time selection methods introduced in this chapter against the baseline, used in Section 3.2 and published in Ben Rached et al. (2020), which used linear CCDF interpolation. In order to make the comparison meaningful, we introduce a modification to the baseline linear CCDF method, to fit it within the same bounded-effort framework as the other two estimators. Since we stipulated bounded total pilot effort we must use a scheme which produces meaningful estimates even where the pilot run suffers premature extinction. We choose a simple *ad hoc* scheme which is that, if at time $t'_k < 1$ the pilot run suffers an extinction (i.e., $n'^{(k)} = 0$) and the run terminates, then we retrospectively update $t'_k \leftarrow 1.01$. This ensures a strictly monotone and thus invertible CCDF on $[0, 1]$, and thence meaningful, if not ideal, time selection. We would expect this construction to be particularly disadvantageous to the linear CCDF method where the sample size is small enough that the extinction probability becomes non-significant.

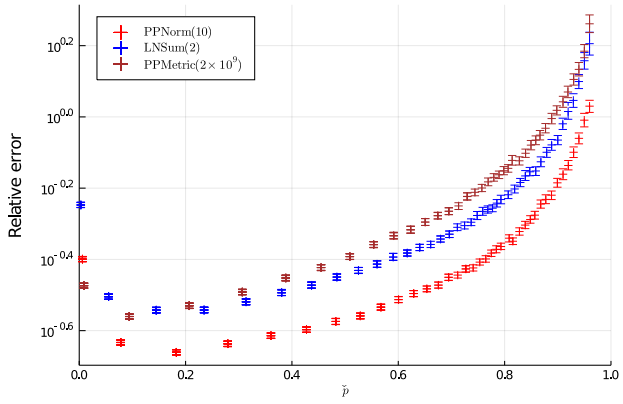
In the Section 3.2 we assumed that it would be reasonable to set η_{pilot} to be "negligible." In practice, in a constrained effort context where it is hard to know what effort is truly negligible. If we have a constrained total effort budget η then effort committed to the pilot run comes at the cost of removing it from the



(a) Small probability target event



(b) Moderate probability target event



(c) Large probability target event

Figure 4.3: Relative error with various values for target survival probabilities $\check{p}$; pilot times $t'_m = m/30$, pilot effort $\eta_{\text{pilot}} = 32 \cdot 10^5$, main effort $\eta_{\text{main}} = 10^3$, $R = 8000$ replicates. Bars denote 95% confidence interval over 1000 bootstrap samples.

main run. As we expect variance to decay approximately as $\mathrm{Var}(\hat{\ell}(\kappa, \eta_{\text{main}})) \propto 1/\eta_{\text{main}}$ the effort removed from the main run can make a material difference to the precision of an estimate — if we were to hold the splitting times constant but simply decrease the effort from η_{pilot} to $\eta - \eta_{\text{pilot}}$, we would expect the relative error of the estimator to grow approximately as

$$\frac{\mathrm{re}(\hat{\ell}(\kappa, \eta - \eta_{\text{pilot}}))}{\mathrm{re}(\hat{\ell}(\kappa, \eta))} \approx \sqrt{\frac{\eta}{\eta - \eta_{\text{pilot}}}}. \tag{4.44}$$

This represents an approximate 'cost' that the pilot run must repay in efficiency of the main run. The trade-off of this effort allocation is not clear. The less effort that is available to the pilot run, the less precise are the estimates of splitting time and thus ultimately the overall estimator quality.

We investigate this trade-off via simulation in the case where total effort is constrained. For fixed total splitting effort budget η we dedicate various proportions of simulation effort to pilot analysis, η_{pilot}/η. To this end, we set the pilot times $t'_k = k/10, k = 1, \ldots, 10$, and pilot per-step population $\tilde{n}' = \lceil \eta_{\text{pilot}} \rceil$. Where $\eta_{\text{pilot}}/\eta = 0$ we take the pilot times to be the main splitting times. There is a choice of parameter K choosing the number of pilot splitting times.

In this setting we are able to perform like-for-like comparison across all the estimators. Figure 4.4 and Figure 4.5 compare performance for the various time selection methods across two different rarity regimes in terms of estimated relative error. In each comparison in this chapter, the *GPD* method denotes that of Section 4.1, *Survival* denotes that of Section 4.2, and the *Linear CCDF* is that of Section 3.2. The wide grey ribbon in each of these denotes the estimated relative error (and its error bars) in the no pilot run scenario, where we simply split at the times $t'_k = k/10$ and dedicate all effort to the main run.

Across these results we see no strictly dominant method for survival time selection. As a rule, dedicating 5-10% of the effort to a simulation run, by either the GPD or survival methods, produces near-optimal results. An exception is the PPNorm(1) case of Figure 4.5b, where no amount of pilot effort allocation to any method beats the no-pilot case. The linear CCDF method is never plausibly the best option. It does beat at least the GPD method in Figure 4.5c. However, the variance of the linear CCDF estimates becomes prone to exploding in the

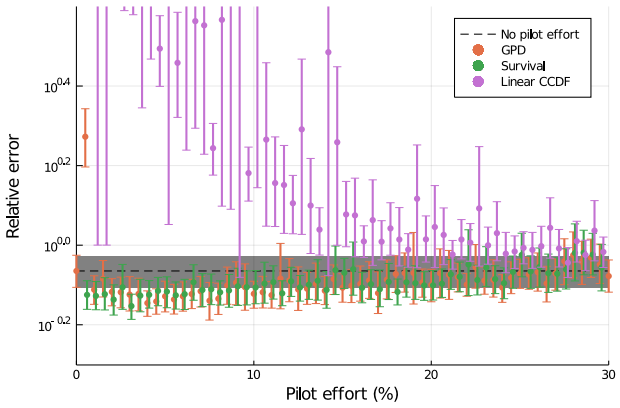
small effort setting. As we have set the estimators to use non-adaptive splitting times in the pilot run, it is not unexpected that eventually it the estimator breaks down. For all the events of medium probability the differences in methods, while significant as measured by confidence intervals, are of small magnitude.

4.3.3 Attainment of target survival probability $\check{p}$

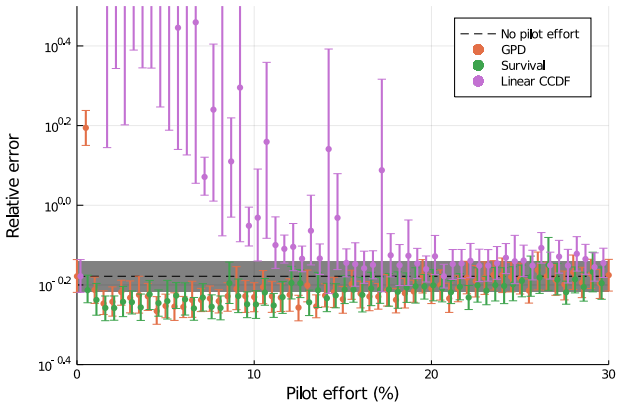
To explore how well we have attained the instrumental goal of enforcing uniform conditional survival probabilities $p^{(m)} = \check{p}$, we visualise the distribution of empirical splitting probabilities achieved in the piloted splitting runs. In order to have a comparable common target splitting time $\check{p}$ we do not squeeze the target probability as per (4.2) but hold $\check{p} = 0.2$ fixed regardless of the pilot estimate $\hat{\ell}'$. Since the final step in this case can, by design, be far from the uniform value $p^{(M)} \neq \check{p}$, we discard all such final steps to avoid introducing extra bias. For the remaining steps, $\check{p}$ is the target value, and we plot that as a dotted line for reference. The empirical $\hat{p}^{(m)}$ values are themselves noisy estimates of the true conditional survival probabilities of a given set of splitting times, so we should expect a non-trivial noise in the empirical realized conditional survival probability estimates even if the splitting times were to be chosen to perfectly produce the desired conditional survival probabilities. Ideally, in a good splitting method the true conditional survival probabilities would be close to the target value.

We do not, of course, have access to the true conditional survival probabilities, but instead must estimate them from empirical survival probabilities in the simulations. In Figure 4.6 and Figure 4.7 we plot the range of empirical survival probabilities for the parameters of the models of Figure 4.4 and Figure 4.5. Specifically, we plot the means and 95% ranges for the concatenation $\hat{p}_r{}^{(m)} = n_r{}^{(m)}/\tilde{n}_r{}^{(m-1)}$, over all steps $m \leq M$ of all replications $r \leq R$ of the main splitting run. We do so for each splitting proportion η_{pilot}/η.

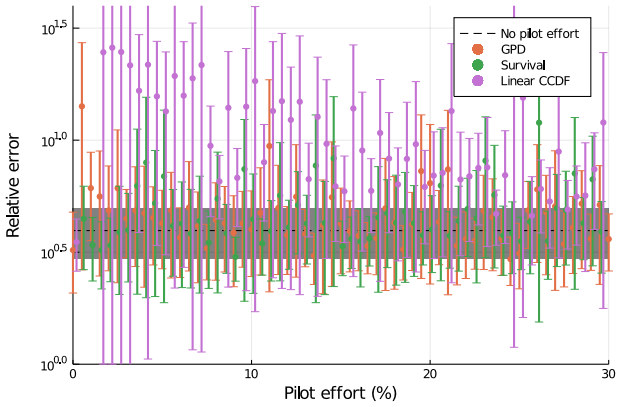
We observe a diversity of behaviours over the different parameter settings. Empirical averages of $\hat{p}^{(m)}$ show signs of being biased away from the target $\check{p}$. It is not clear that the times attained are consistent in increasing pilot effort. The magnitude and direction of the bias depends on both model and splitting method. Asymptotically we observe that the range of empirical conditional survival prob-



(a) LNSum(0.02)

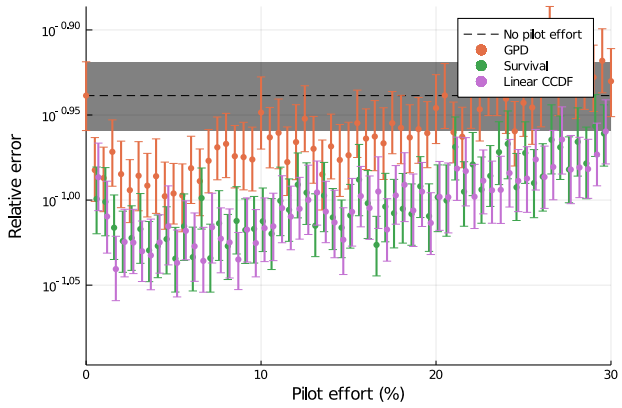


(b) PPNorm(0.1)

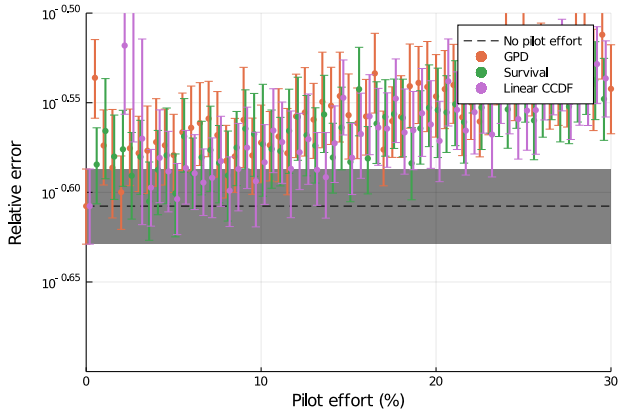


(c) PPMetric($2 \cdot 10^7$)

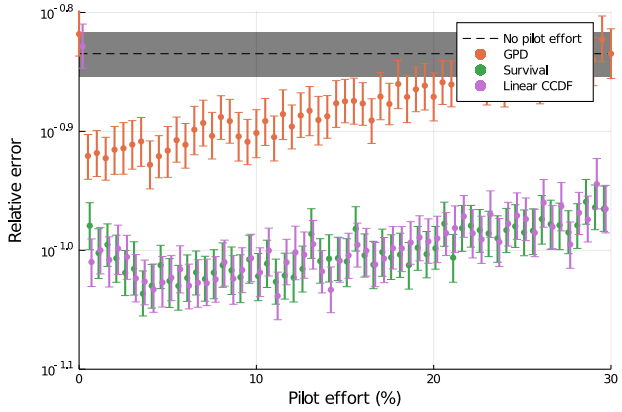
Figure 4.4: Pilot effort proportion and relative for a very rare event, pilot times $t'_k = k/10$, total effort $\eta = 10^4$, $R = 10^3$ replications. Bars denote 95% confidence interval over 1000 bootstrap samples. Series are offset horizontally for legibility.



(a) LNSum(0.2)



(b) PPNorm(1)



(c) PPMetric($2 \cdot 10^8$)

Figure 4.5: Pilot effort proportion and relative error for a somewhat rare event, pilot times $t'_k = k/10$, total effort $\eta = 10^4$, $R = 10^3$ replications. Bars denote 95% confidence interval over 1000 bootstrap samples. Series are offset horizontally for legibility.

abilities in the survival method is typically smaller than the GPD, although the GPD method can outperform it in certain low-pilot-effort regimes. Either the linear CCDF or the survival estimate can attain a closer mean cumulative survival probability to the target, depending upon the problem. In short, the differing problem structures appear to lead to different ideal choices of optimal splitting time selection methods, at least as regard the range and expectation of attain conditional survival probabilities. We see that in practice while the precision *can* improve over the baseline with regard to optimal splitting time selection, improvement in this regard is not necessarily a given.

4.3.4 Large-effort asymptotics of the combined estimator

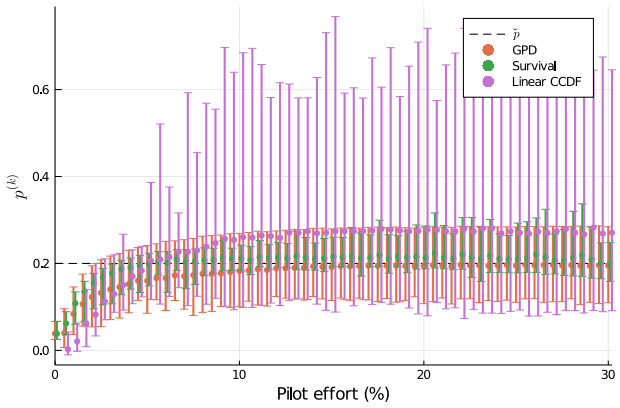
The next numerical simulations examine the efficiency of the various simulation methods, applied to the various problems, in the large-effort limit.

For all combinations of model problems and of estimators, we estimate the the empirical ENRV as a general indicator of the quality of the combination. For each combination we choose several different effort levels and monitor the large-effort scaling behaviour of the estimator. Pilot effort for each is fixed at $\eta_{\text{pilot}}/\eta = 0.05$ with $t'_k = k/20$ except for the *No pilot* case, where we fix $\eta_{\text{pilot}} = 0$ and set the main splitting times to the same, $t_m = m/20$. We estimate the large-sample asymptotic limit by weighted means of the ENRV estimates. Writing $\text{ENRV}(\hat{\ell}(\infty)) \stackrel{\text{def}}{=} \lim_{\eta\to\infty} \text{ENRV}(\hat{\ell}(\eta))$, the weighted estimator is

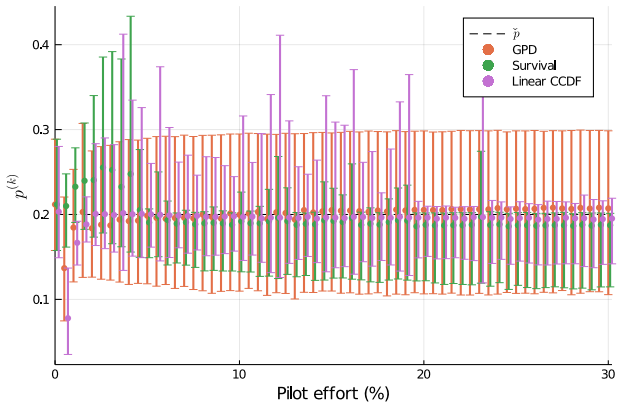
$$\widehat{\text{ENRV}}(\hat{\ell}(\infty)) \stackrel{\text{def}}{=} \frac{1}{\sum_\eta w_\eta} \sum_\eta w_\eta \widehat{\text{ENRV}}(\hat{\ell}(\eta)). \tag{4.45}$$

We set the weights to be inverse variances, $w_\eta = 1/\widehat{\text{Var}}[\widehat{\text{ENRV}}\hat{\ell}(\eta)]$, where the variances in question are estimated by bootstrap resampling. In general, as the effort increases from low levels there is a transient stage wherein the ENRV has a high variance and may be far from an apparent asymptotic limit. We recall that we only expect the ENRV to be asymptotically constant (2.30), so this behaviour is not entirely surprising.

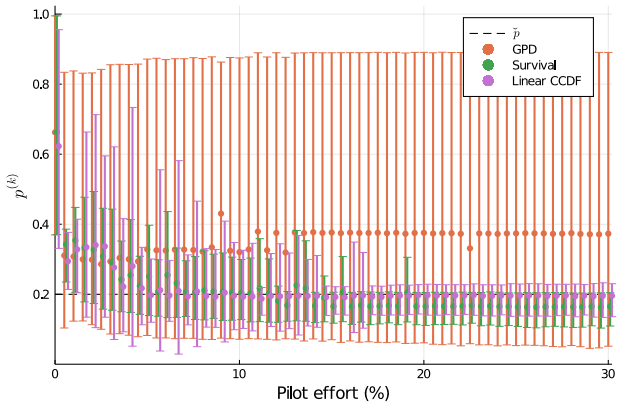
Across the different example problems we observe substantively different behaviour of the combined quasi-monotone splitting estimator. Firstly we consider



(a) LNSum(0.02)

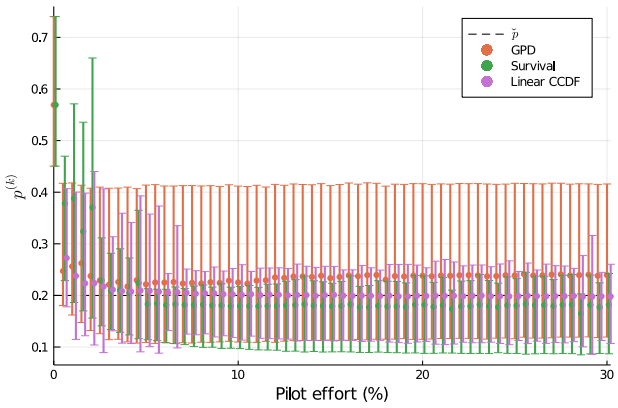


(b) PPNorm(0.1)

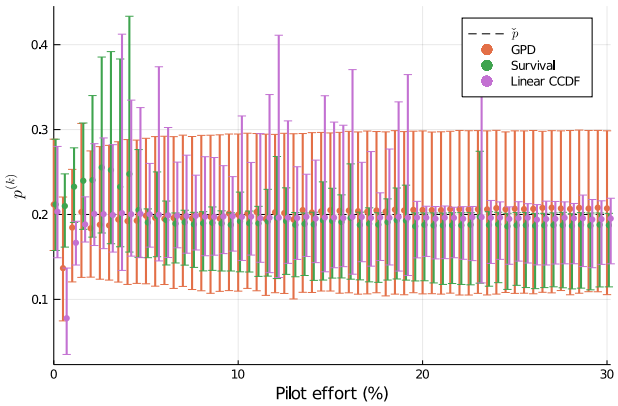


(c) PPMetric($2 \cdot 10^7$)

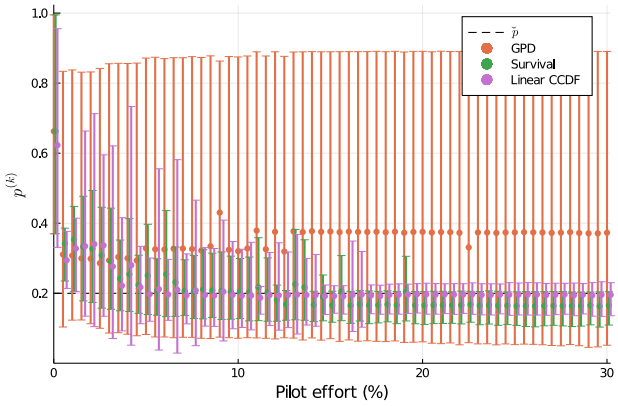
Figure 4.6: Pilot effort proportion and attained $\check{p}$ for a very rare event, pilot times $t'_k = k/10$, total effort $\eta = 10^4$, $R = 10^3$ replications. Central mark denotes mean and error bars denote 95% range. Series are offset horizontally for legibility.



(a) LNSum(0.2)



(b) PPNorm(1)



(c) PPMetric($2 \cdot 10^8$)

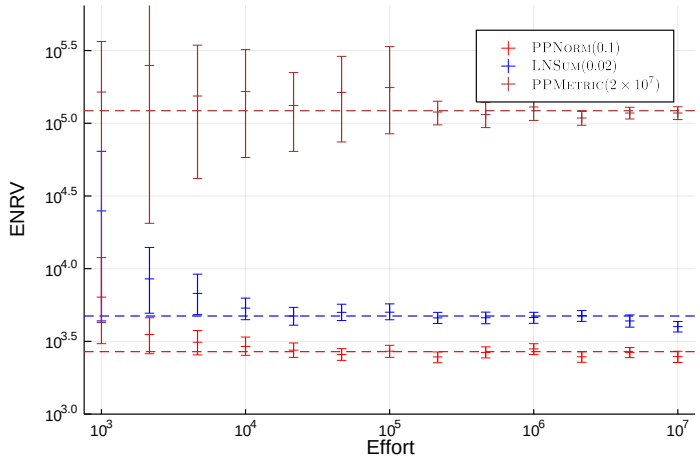
Figure 4.7: Pilot effort proportion and attained $\check{p}$ for a somewhat rare event, pilot times $t'_k = k/10$, total effort $\eta = 10^4$, $R = 10^3$ replications. Central mark denotes mean and error bars denote 95% range. Series are offset horizontally for legibility.

the different behaviour of each of the model problems,plotted in Figure 4.8. For these simulations we fix all parameters apart from effort, and choose κ values such that the target event probabilities are comparable, with $\ell \approx 5 \cdot 10^{-15}$. Specifically, $\hat{\ell}(\text{PPNorm}(0.1)) \approx 7.932 \cdot 10^{-15}$, $\hat{\ell}(\text{LNSum}(0.02)) \approx 4.671 \cdot 10^{-15}$, and $\hat{\ell}(\text{PPMetric}(2 \cdot 10^7)) \approx 5.724 \cdot 10^{-15}$. The central limit theorem for fixed effort splitting (Theorem 2.1) additionally tells us that estimates from splitting methods with fixed intermediate target sets are eventually normally distributed; we know of no such results where, as here, the intermediate target sets are estimated adaptively by a pilot run, although numerically we see that asymptotic normality is a plausible hypothesis. We investigate the asymptotic distribution numerically by applying the Anderson Darling test of the null hypothesis that samples are drawn from a normal distribution. The results of this experiment are plotted in Figure 4.8b. All three models show an increasing tendency to fail to reject the null as effort increases, indicating a distribution that is in the sense of this test at least, more normally distributed. The rate of convergence is problem dependent. The example of PPMetric is particularly slow to converge. Even at $\eta = 10^6$, the estimate is not plausibly normally distributed.

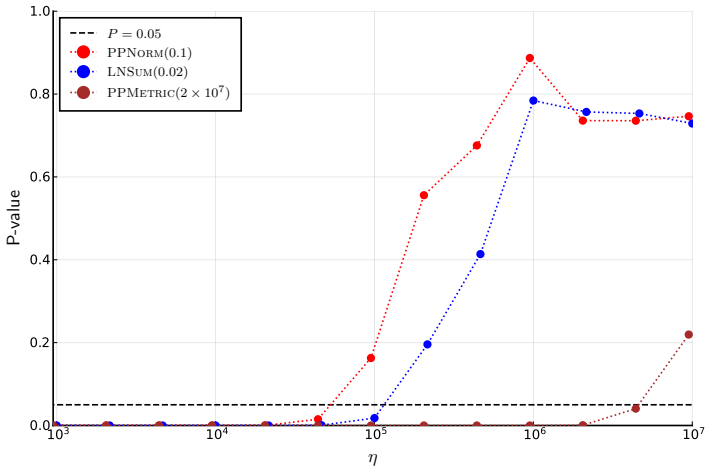
Holding the problem fixed and varying the time selection methods, results for the PPNorm(0.1) and PPMetric(2×10^7) problem are plotted in Figure 4.9 and Figure 4.9, and tabulated in Table 4.2 and Table 4.3. Results for other model is omitted, since it demonstrates no new behaviour. As the effort changes, the ranking of the preferred time selection method changes, and none is strictly dominant over the whole range. Asymptotically, we notice that the GPD method performs best, at least measured by having the smallest point estimate of asymptotic ENRV. This is and inversion of the order we found in the small effort regime, where the GPD method's performance was unspectacular. Note, however that the overall relative errors in question are small at this high effort level.

4.3.5 Small probability asymptotics of combined estimator

We consider also whether the quasi-monotone splitting method has attained logarithmic relative efficiency (2.19). We recall the estimation method of Section 3.4, in which estimated the closeness to logarithmic relative efficiency by examining the



(a) ENRV of splitting estimates. Error bars shows 95% confidence interval over 1000 bootstrap replicates. Dashed line denotes inverse-variance-weighted sample mean $\widehat{\mathrm{ENRV}}(\hat{\ell}(\infty))$.



(b) P-value for the Anderson-Darling test of the hypothesis that samples are drawn from a normal distribution. Dotted lines are a visual aid only.

Figure 4.8: Effort normalized relative variance of quasi-monotone estimators, over $R = 1000$ realizations of the estimator for various problems. Splitting times are found by the survival method with $\eta_{\text{pilot}}/\eta = 0.05$.

Table 4.2: Estimated large sample efficiency of quasi-monotone estimators, in PPNorm(0.1) over $R = 100$ realizations of the estimator for each time selection method.

Method	$\widehat{\mathrm{ENRV}}(\hat{\ell}(\infty))$
No Pilot	$3.80 \cdot 10^3$
GPD	$2.58 \cdot 10^3$
Survival	$2.85 \cdot 10^3$
Linear CCDF	$3.51 \cdot 10^3$

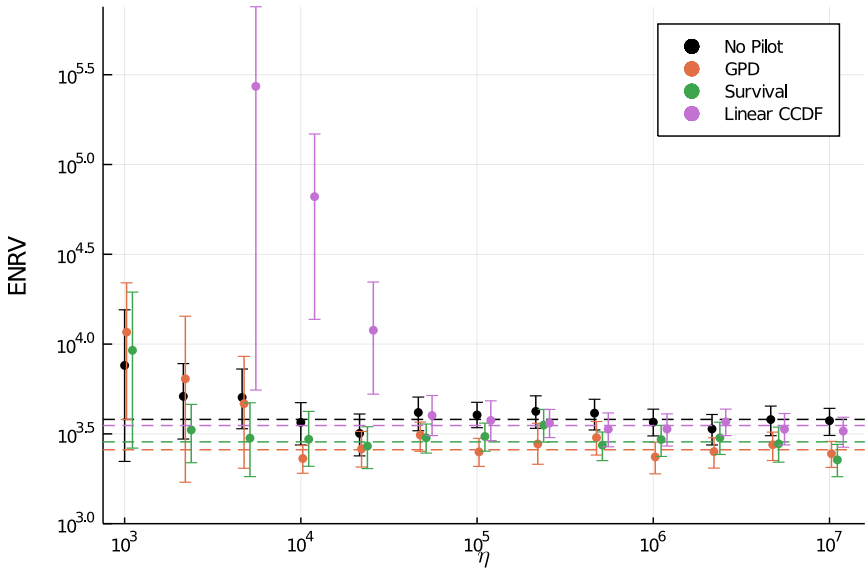


Figure 4.9: Estimated large sample efficiency of quasi-monotone estimators, over $R = 100$ realizations of the estimator for various time selection methods in PPNorm(0.1). Pilot effort satisfies $\eta_{\mathrm{pilot}}/\eta = 0.05$ with $t'_k = k/20$. Series are offset horizontally for legibility. Error bars show bootstrap 95% confidence intervals over 1000 repetitions.

Table 4.3: Estimated large sample efficiency of quasi-monotone estimators, in PPMetric(2×10^7) over $R = 100$ realizations of the estimator for each time selection method.

Method	$\widehat{\mathrm{ENRV}}(\hat{\ell}(\infty))$
No Pilot	$1.35 \cdot 10^5$
GPD	$1.71 \cdot 10^5$
Survival	$1.23 \cdot 10^5$
Linear CCDF	$1.55 \cdot 10^5$

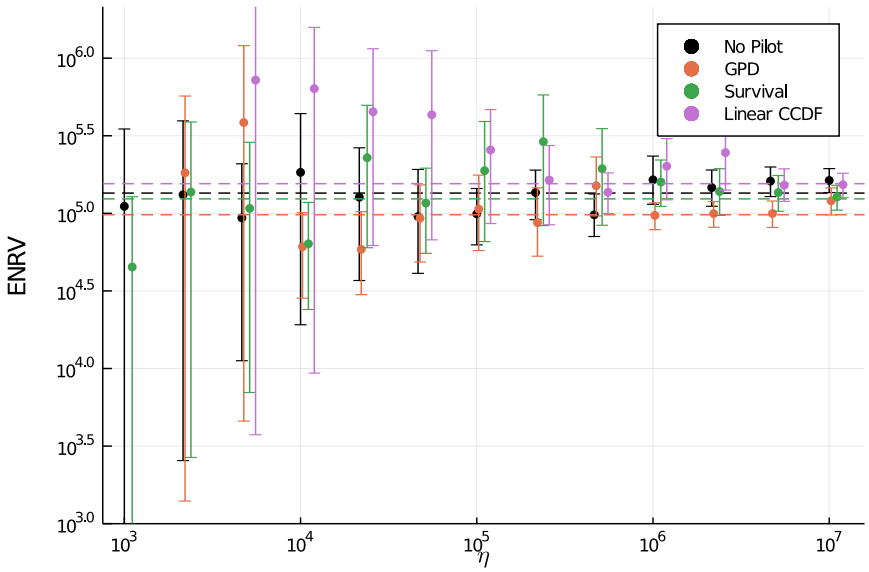
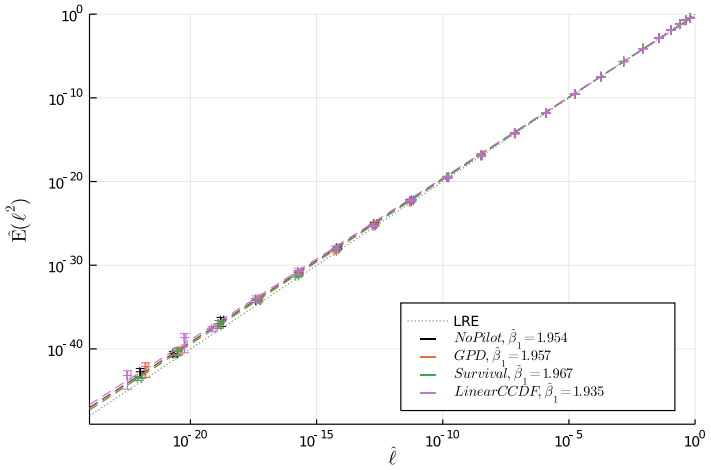


Figure 4.10: Estimated large sample efficiency of quasi-monotone estimators, over $R = 100$ realizations of the estimator for various time selection methods in PPMetric(2×10^7). Pilot effort satisfies $\eta_{\text{pilot}}/\eta = 0.05$ with $t'_k = k/20$. Series are offset horizontally for legibility. Error bars show bootstrap 95% confidence intervals over 1000 repetitions.

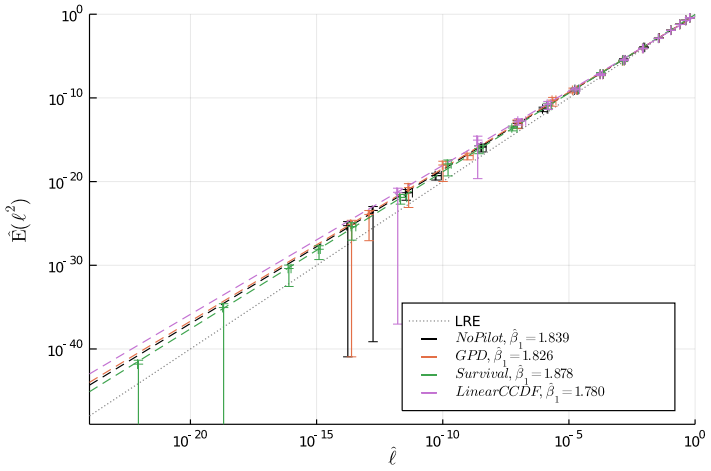
Table 4.4: Small-probability-asymptotic behaviour in quasi-monotone splitting PPMetric under the various time selection methods. Regression confidence intervals have no straightforward interpretation and are are not reported.

η	Method	$\hat{\beta}_1$
10^3	No Pilot	1.839
10^3	GPD	1.826
10^3	Survival	1.878
10^3	Linear CCDF	1.780
10^5	No Pilot	1.953
10^5	GPD	1.957
10^5	Survival	1.967
10^5	Linear CCDF	1.935

slope of the regression line of $\log \mathbb{E}[\ell^2]$ against $\log \hat{\ell}$. Applying the same method here, the results for PPMetric are plotted in Figure 4.11. The linear regression fit is compatible with the hypothesis that we approach logarithmic efficiency with higher efforts, but do not do so for smaller efforts. In the higher effort case, plotted in Figure 4.11a, we achieve respectable values for the GPD and survival methods, with a coefficient $\hat{\beta}_1 > 1.95$ for all methods. If we reduce the effort to $\eta = 10^3$ as in Figure 4.11b we are somewhat further from logarithmic efficiency, attaining coefficients in the range $1.78 - 1.82$. We do generally better with survival-based splitting time estimates than with the other methods, in that survival-method pilot runs consistently exceed the baseline without a pilot run. The observations here are compatible with the interpretation that we are "somewhat close" to logarithmic relative efficiency, and that our time selection method enhancements have improved at least in some cases of small-probability efficiency. A similar pattern, with slightly different coefficient estimates, is reproduced across the other example problems. Since the results are not qualitatively different results so are not shown. The caveat here is that we do not have good confidence bounds for $\hat{\beta}_1$ as an estimator of true logarithmic efficiency, the estimates are suggestive only.



(a) $\eta = 10^5$.



(b) $\eta = 10^3$.

Figure 4.11: Estimated small-probability-asymptotic efficiency in quasi-monotone splitting PPMetric$(2 \cdot 10^7)$ under the various time selection methods. Dashes show lines of best fit. Error bars show 95% bootstrap confidence interval

4.4 Guidance for practitioners

This simulation study has can be summarised as supporting that in the extremely small effort regime, all methods are fragile and noisy. For higher effort levels, the survival methods with a small pilot effort fraction frequently, although not universally, improve the overall estimator variance. At high effort levels we are generally indifferent to the choice of pilot run method. This heuristic advice leaves details undetermined. In practice, the choice of the tuning parameters $t'_1, \ldots, t'_K$ and η_{pilot}/η for a given application must depend upon the details of that application. A reasonable rule of thumb based on these simulation studies is to dedicate $\eta_{\text{pilot}}/\eta = 5 - 10\%$ of the total effort to a splitting run using the survival analysis method. Where the target event is particularly common and the effort budget high, this pilot budget could possibly be reduced.

Choice of pilot $t'_1, \ldots, t'_K$ in this study was, as noted, arbitrary. Heuristically, if we tolerate a survival probability of 10% at each step, the value used in this chapter, $K = 20$, would be sufficient to encompass events with probability down to $\ell \simeq 10^{-20}$, smaller than the values we use here. If this does not seem sufficiently robust, outside of the constraint of this chapter that we should be able to compare like-effort for like-effort, using some kind of adaptive splitting for the pilot run could provide an automatic means of selecting $t'_1, \ldots, t'_K$.

We have faced here a challenge characteristic of much fixed-effort splitting research, which is that it is hard to produce analytic guarantees about the behaviour of our estimators. Our solution has been to apply various statistical models (weighted means, linear regression) and tests (Anderson-Darling) to understand the distribution of estimators in simulation studies. The upshot has been compatible with certain hypotheses — that the hybrid estimators are not far from logarithmically efficient, that the proposed time-selection procedures can reduce estimator error, and that the resulting estimators are asymptotically normally distributed. All of these are to differing degrees dependent upon the level of effort, the rarity of the target event, and the specifics of the problem in question.

On the basis of these hypotheses we have formulated heuristic recommendations about pilot effort and time selection methods. The universality of these recommendations is subject to the usual limitations that apply to simulation-based

studies, and which are motivated by idealized and asymptotic analysis. We are on especially shaky ground in the low-effort context, i.e. $\eta < 10^4$, and in this domain recommendations are unclear. With higher effort levels, pilot runs using survival-analysis-based time selection methods with a 5% effort allocation seems a low-risk choice in most cases. In trials, this improves over the level-selection method of Section 3.2 in intermediate effort levels, at a minimal cost in efficiency compared to the case without a pilot run.

Chapter 5

Conclusions: Quasi-monotone splitting

In Chapter 3 we introduced a type of splitting method, the quasi-monotone splitting sampler. This comes equipped with a convenient, simple procedure to design splitting samplers across a wide variety of problems with appropriate structure. The resulting samplers have the form of a Sequential Monte Carlo algorithm which simulates and resamples realizations over successive intervals of a synthetic Markov process whose terminal distribution is our distribution of interest, i.e., the dynamic splitting for static problems technique. Simulating the intermediate sets using the quasi-monotone splitting method is straightforward, since we only need to inspect the values of the process at the final instant of each interval, which leads to simple and largely automatic simulation algorithms. The problem of selecting time instants, we have argued though idealization arguments, can be approached by studying the lifetime distribution of particles in the sampler. Estimating these particles' survival time quantiles gives us uniform conditional survival probabilities $\check{p} = p^{(1)} = \cdots = p^{(1)}$. In Chapter 4 we introduced tools from survival analysis and extreme value theory to study these lifetime distributions. This provided us means, based on the output of a pilot run, to estimate the optimal splitting times which approach these uniform conditional survival probabilities. Throughout this process, no splitting time method strictly dominated in terms of our relative error metrics. We have provided heuristic advice for selecting methods based on

numerical simulation of their effectiveness.

It is tempting to consider improvements to the method of selecting the splitting times. Although we are able to improve the variance of the overall splitting method, it is not clear that the optimal times themselves are consistently estimated by any of our methods across all the example problems. Moreover, in the large effort limit, pilot run effort can become counter-productive. Many restrictive design choices were made the statistical construct these estimators, and we might suppose that lifting these restrictions could improve the behaviour of the combined estimator. For example in the lifetime distribution estimators, we can construct more sophisticated mixture models hybridising the nonparametric survival analysis and GPD estimators, as seen in applied extreme value theory models (Markovitch and Krieger 2002; McNeil 1997). We can also use data-driven methods to select parameters of the somewhat arbitrary regularization we used to construct the GPD and survival estimators. Elaborations of our weighting scheme, such as learning from data the weighting function w in iteratively reweighted model fits, are also possible. We might also suppose that our estimators could be improved by directly estimating the quantile (3.29) of interest, directly estimating $t_m = \bar{T}^{-1}(\check{p}^m)$, rather than estimating the CCDF $\bar{T}$, which is essentially a nuisance parameter as regards choosing the times. Direct estimation of quantiles in an Extreme Value Theory context is considered in, e.g. Bhatti et al. (2018), Hosking and Wallis (1987), and Makarov (2006). The use of splitting methods to directly estimate quantiles is explored in (Guyader, Hengartner, and Matzner-Løber 2011). We can also suppose that switching between linear, survival and GPD CCDF estimators based on the behaviour of the estimator might improve accuracy.

Such extensions are likely to be feasible. Whether they are worthwhile is less certain. The magnitude of the improvement in the overall estimator from improved level selection has been only modest in terms of improvement in the accuracy of the splitting estimates. Our objective of uniform conditional survival probabilities is in any case merely a surrogate for the true objective of minimising estimator error. Having proceeded a way down this path of adjusting t_m values to more closely approximate the surrogate objective, it is unclear that further pursuing the means to better approximate this surrogate objective 'plucks the lowest hanging fruit'. The idealizations for the level selection which we use to justify our methods

include many assumptions. Possibly most notably, we made certain large-effort asymptotic approximations of which we are particularly suspicious in the small-effort regime where efficiency is most crucial. Further, layering complexity upon complexity in a method that we have espoused for its simplicity could undermine that desirable simplicity.

An alternative strategy could discard surrogate objectives in favour of direct optimization of true objectives, or at least a closer surrogate. Directly estimating and optimising the error distribution of parametric Monte Carlo samplers is a large, active field in the study of Monte Carlo methods. Various authors have explored directly optimising sampling distributions by minimising a loss function (see sumary in Robert et al. 2018) and there is reason to suppose it could be feasible in quasi-monotone splitting. Here, approximate gradient information for sampling parameters with respect to some measure of sampling efficiency would be available by automatic differentiation (Mohamed et al. 2020). The virtue of such an approach is that it potentially enables us to optimise a broader selection of parameters than simply a selection of splitting times. Using gradients we could exploit information in particle state values rather than using only population counts as in the idealized problem. In such a setup, we might feasibly optimise ideal parameters for a differentiable parametric importance function S_g. We could optimise choice of time instants over a multivariate time axis, or parameters of the latent processes themselves. This would align with the modern enthusiasm for methods using gradient-based optimization to improve convergence in sampling problems (e.g. Caterini, Doucet, and Sejdinovic 2018; Salimans, Kingma, and Welling 2015).

A largely orthogonal question to this is whether we can avoid the additional parameter choices for pilot runs by using an entirely adaptive splitting strategy, where a single adaptive splitting run calculates both optimal splitting parameters and the desired target sample simultaneously. There is an active literature on adaptive methods for splitting, encompassing methods more sophisticated than our *ad hoc* adaptive pilot method of Section 3.2, (e.g. Bréhier, Goudenège, and Tudela 2016; Bréhier, Lelièvre, and Rousset 2015; Cérou and Guyader 2007; Cérou and Guyader 2016; Charles-Edouard et al. 2015). Adaptive methods pay a cost in increased efficiency and complexity of analysis, so the utility of such a trade-off is not immediate; this is once again a subject for future research.

These speculations aside, the quasi-monotone splitting sampler is available to use as-is, attains good performance on many problems, has broad generality and a simple implementation. In many cases this easy procedure turns out to produce state-of-the-art results. In the quasi-monotone setting, the task of extending the method to novel distributions is a largely mechanical process, which we have illustrated with several examples. Many more can be easily constructed by users of the algorithm. The methods developed here suggest suggest, further, the potential for new hybridization with other splitting methods. Together, these factors lead us to argue that this work advances the frontier of rare event Monte Carlo methods.

Appendix A

Selected univariate distributions

A.1 Poisson distribution

The Poisson distribution is a univariate distribution intimately related to the Poisson process.

Definition A.1 (Poisson distribution)**.** A random variate G supported on $\mathbb{N}^0$ with probability mass function $\mathbb{P}[G = k; \lambda]$ is Poisson distributed if that mass function is

$$\mathbb{P}[G = k; \lambda] = \frac{\lambda^k \mathrm{e}^{-\lambda}}{k!} \mathbb{I}\{k \in 0, 1, 2, \dots\}. \tag{A.1}$$

We write $G \sim \text{Poisson}(\lambda)$. We refer to λ as the *rate*.

If $G \sim \text{Poisson}(\lambda)$ then

$$\mathbb{E}[G] = \lambda \tag{A.2}$$

$$\text{Var}[G] = \lambda. \tag{A.3}$$

We use the following well-known facts about the Poisson distribution. Suppose $G \sim \text{Poisson}(\lambda)$, $G' \sim \text{Poisson}(\lambda')$ and $G \perp\!\!\!\perp G'$. Then

$$G + G' \sim \text{Poisson}(\lambda + \lambda'). \tag{A.4}$$

From this it follows that the Poisson distribution is *infinitely divisible*, by which we mean that any Poisson distributed random variate is equivalent in law to a sum

of differently-parameterized independent Poisson random variables. We use this property extensively when discussing the Poisson process.

A.2 Gamma distribution

The gamma distribution is a univariate distribution supported on the non-negative reals. We use it here to find the increments of the gamma process.

Definition A.2 (Gamma distribution)**.** A random variate G with density $g(x; \alpha, \lambda)$ is gamma distributed if that density is

$$g(x; \alpha, \lambda) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha - 1} \mathrm{e}^{-\lambda x}, x \geq 0. \tag{A.5}$$

We write $G \sim \text{Gamma}(\alpha, \lambda)$. We refer to λ as the *rate* and α as the *shape*

If $G \sim \text{Gamma}(\alpha, \lambda)$ then

$$\mathbb{E}(G) = \alpha/\lambda \tag{A.6}$$

and

$$\text{Var}(G) = \alpha/\lambda^2. \tag{A.7}$$

In particular,

$$\text{Gamma}(1, \lambda) \overset{\text{D}}{=} \text{Exp}(\lambda). \tag{A.8}$$

We use various standard facts about the gamma distribution (Applebaum 2009; Asmussen and Glynn 2007; Rubinstein and Kroese 2016).

Theorem A.1 (Transformations of gamma variates)**.** Suppose $G \sim \text{Gamma}(\alpha, \lambda)$, $G' \sim \text{Gamma}(\alpha', \lambda)$, $G \perp\!\!\!\perp G'$, and $c \neq 0$ is a real-valued constant. Then

1. $G + G' \sim \text{Gamma}(\alpha + \alpha', \lambda)$ (superposition)
2. $cG \sim \text{Gamma}(\alpha, \lambda/c)$ (scaling)

From theorem B.1.1 it follows that the gamma distribution is, like the Poisson, infinitely divisible and thus a gamma distributed random variate is equivalent in law to a sum of differently-parameterized independent gamma random variables.

Appendix B

Subordinators

Subordinators are the predominant class of stochastic process used in quasi-monotone splitting. These are pathwise almost-surely non-decreasing Lévy processes, which are themselves a useful category of Markov processes. We introduce Lévy processes generally first, then specialise to subordinators in particular, and then the particular subordinators used in our models.

B.1 Lévy processes

Lévy processes are the most general class of $\mathbb{R}^d$-valued processes with independent, stationary increments. For a deeper presentation, see e.g. Bertoin (1996), Kyprianou (2014), and Sato (1999).

Definition B.1 (Lévy process)**.** A d-dimensional Lévy process $\{G(t)\}_{t\in I}$ is a stochastic process indexed by an interval $I \subseteq \mathbb{R}$ and taking values in $\mathbb{R}^d$, such that it possesses

1. Independent increments: $G(t) - G(s)$ is independent of $\{G(u) \colon u \leq s\}$ for any $s < t$.

2. Stationary increments: $G(s+t) - G(s)$ has the same distribution as $G(t) - G(0)$ for any $s, t > 0$.

3. Continuity in probability: $G(s) \to G(t)$ in probability as $s \to t$.

We deduce Lévy processes are Markov, by 1 and 2.

We often discuss Lévy processes through their increment distribution, which specifies all finite dimensional distributions of the process with respect to the natural filtration. Presuming we may simulate from the increment distribution, we may simulate points on a path of a Lévy process by summing increments over increasing time steps.

$$G(t_i) \simeq \sum_{j<i} (G(t_{i+1} \mid G(t_i)) - G(t_i)) = \sum_{j<i} G_j. \tag{B.1}$$

Definition B.2 (Subordinator)**.** A Lévy process $\{G(t)\}_{t\in I}$ is a subordinator if the increment distribution is, coordinate-wise, non-negative, i.e., for $t > s$

$$\mathbb{P}[G(t) - G(s) \geq \mathbf{0}] = 1 \tag{B.2}$$

where the inequality here is interpreted coordinate-wise in the case of vector processes.

Properties of particular subordinators, such as the gamma (Section B.2) and Poisson processes (Section B.3), are introduced here.

B.2 Gamma process

A *gamma process* is a Lévy process whose increments are gamma distributed (Ferguson 1974; Ferguson and Klass 1972). Gamma processes comprise a sub-class of subordinators that arises naturally in a variety of applications (Applebaum 2009; Asmussen and Glynn 2007; Rubinstein and Kroese 2016). We review some properties here.

Definition B.3. A univariate Lévy process $\{G(t; \alpha; \lambda\}_t$ is a *gamma process* if for any index values $t \geq s$ increments of the process are distributed

$$G(t; \alpha, \lambda) - G(s; \alpha, \lambda) \sim \text{Gamma}(\alpha(t-s), \lambda). \tag{B.3}$$

We write $\{G(t)\}_{t\in[0,\infty]} \sim \text{GammaProc}(t; \alpha, \lambda)$. Where the index argument t is unambiguous we suppress it, writing $\text{GammaProc}(\alpha, \lambda)$.

This corresponds to increments per unit time in terms of $\mathbb{E}(G(1)) = \alpha/\lambda$ and $\mathrm{Var}(G(1)) = \alpha/\lambda^2$. It follows that $G(1;\alpha,\lambda) \sim \mathrm{Gamma}(1,\lambda t) = \mathrm{Exp}(\lambda)$.

Some rules for gamma processes follow from the analogous results for the Gamma distribution.

Proposition B.1 (Transformations of gamma processes)**.** Suppose $G = \{G(t)\}_{t\in[0,1]} \sim \mathrm{GammaProc}(\alpha,\lambda)$, $G' = \{G'(t)\}_{t\in[0,1]} \sim \mathrm{GammaProc}(\alpha',\lambda)$, $G \perp\!\!\!\perp G'$, and $c \neq 0$ is a scalar constant. Then,

1. $G + G' \sim \mathrm{GammaProc}(\alpha+\alpha',\lambda)$ (superposition)
2. $\{cG(t;\alpha,\lambda)\} \overset{\mathrm{D}}{=} \{G(t;\alpha,\lambda/c)\} \sim \mathrm{GammaProc}(\alpha,\lambda/c)$ (scaling)
3. $\{G(ct;\alpha,\lambda)\} \overset{\mathrm{D}}{=} \{G(t;c\alpha,\lambda)\} \sim \mathrm{GammaProc}(c\alpha,\lambda)$ (dilation)

Proof. Part 1 follows from theorem A.1.1. Part 2 follows from theorem A.1.2. Part 3 follows by substituting ct for t in (B.3) and comparing with (A.5). □

A d-dimensional gamma process is the concatenation of d mutually-independent univariate Gamma processes into a d-dimensional vector.

B.3 Poisson process

We call a Lévy process $\{G\}_t \sim \mathrm{PoissonProc}(\lambda)$ is a *Poisson* process if, in addition to the usual Lévy process requirements, the increments of the process have a Poisson distribution. Specifically, for $t_j \geq t_i$:

$$G(t_j) - G(t_i) \sim \mathrm{Poisson}\left((t_j - t_i)\lambda\right). \tag{B.4}$$

Equivalently, it is a counting process whose inter-jump times $\{t_i\}_{i\in G^+}$ are identically and independently distributed such that $t_i - t_{i-1} \sim \mathrm{Exp}(1/\lambda)$, and each jump is of size 1 a.s. We set all $t_0 = 0$ and $G(0) = 0$ a.s. $G : \mathbb{R} \mapsto \mathbb{Z}^+$ such that $G(t) \equiv \sum_{i=1}^{G} \mathbb{I}_{\{t_i<t\}}$. It is easy to show that $G(t) \sim \mathrm{Poisson}(\lambda t)$ (A.1).

Note also the standard result that

$$\lambda \overset{\mathrm{def}}{=} \lim_{h\to 0} \frac{\mathbb{E}\left(G(t,t+h)\right)}{h}. \tag{B.5}$$

We call λ the *rate*.

Part II

Autocorrelogram mosaicing

In this part we introduce a different project whose central object is audio signal processing. Our particular context is a style transfer task. Problems such as these are of great industrial interest, both in speech synthesizers for voice-based computer interactions and in musical and creative audio applications.

This part is based on the work for the conference paper MacKinlay and Botev (2019) of which Dan MacKinlay is primary author. Permissible minor corrections and revisions for thesis style have been made, but the body of the text remains substantively unaltered in line with the declaration of published works.

We make our Python code openly available for public use.[1] This code release is significant, in that at time of writing this was apparently the only openly-released code available for this task in an open-source programming language, and the only one that can be executed without substantial difficulty in forensic reconstruction of an obsolete code environment. Benchmark code in this domain, such as NiMFKS, which is our major point of comparison here, is implemented in MATLAB, which is not only expensive, but prone to sensitive version dependencies; many of the openly available MATLAB options such as MATConcat (Sturm 2004) or Coleman's descriptor-driven synthesis (Collins and Sturm 2011) have not been recently maintained and present considerable difficulties for the user to using recent MATLAB installations. We hope that by lowering the barrier for entry, we might improve the reproducibility of results in this area.

[1] https://github.com/danmackinlay/mosaicing_omp_ismir_2019/

Chapter 6

Mosaic synthesis background

We explain the background to audio style transfer problems generally and the mosaicing problem specifically.

Mosaicing synthesis is a particular approach to the style transfer problem. An important task here is *style transfer* which attempts to synthesize a new signal from two others, a *source* and a *target*. The new synthetic signal should possess the microscopic "stylistic" statistics of the source, and the macroscopic "semantic" statistics of the target. In musical applications the style is usually musical timbre, and the content is the melody and rhythm of the performance. Concretely, if the target were a trumpet playing a melody, and the source a recording of a singing vocalist, the mosaic might aim to emulate the vocalist singing that melody.

There exist a variety of problem definitions of, and associated algorithms for, mosaicing synthesis; (e.g. Caetano and Rodet 2013; Coleman, Maestre, and Bonada 2010; Collins and Sturm 2011; Hoffman and Cook 2006; Hoffman, Cook, and Blei 2009; Simon et al. 2005; Sturm 2004; Sturm et al. 2009; Zils and Pachet 2001), partially summarized in Schwarz (2011). In mosaicing specifically, we accomplish style transfer using a dictionary-based granular synthesis method. Such methods construct their output by superposition of transformed short recordings, *grains*, from an audio dictionary. This superposition is traditionally conducted in the time domain as in classic granular synthesis (e.g. Driedger and Müller 2016; Verhelst and Roelands 1993). More recently spectral domain methods have become popular (Aarabi and Peeters 2018; Buch, Quinton, and Sturm 2017; Driedger and

Pratzlich 2015). The latter methods work on superposing power spectral densities of signals and then estimating phases by some kind of phase retrieval algorithm (e.g. Driedger and Müller 2016; Griffin and Lim 1984; Perraudin, Balazs, and Sondergaard 2013).

Granular synthesis methods in themselves are well understood and widely deployed in industrial applications. They comprise a significant proportion of the music industry market for software synthesizers, are integrated into every major Digital Audio Workstation package, and have been extensively researched (e.g. Roads 2004, and references therein).

The extension of granular synthesis into a style-transfer problem as mosaicing is less well-understood. In this setting we choose the parameters of a granular synthesis so as to optimally approximate a desired target audio signal in the sense of optimising some measure of acoustic similarity. Typically this implies approximating, in the sense of minimising some approximation objective, the power spectral density (PSD) of the target signal. Applications for this include musical accompaniment, creative musical effects, or user customization of speech synthesis (Chazan and Hoory 2006).

Our sparse autocorrelogram method advances the capabilities of musical mosaicing applications, by leveraging a feature map that is related to, but more convenient than, classical PSD methods. This method is enabled by two innovations.

Firstly, we define signal similarity through the *autocorrelogram*, a representation of a time-domain signal in terms of covariance with delayed copies of itself. The autocorrelogram and its relationship to PSD is well-known (e.g. Wiener 1930) but our use in mosaicing synthesis appears novel. Although we use the autocorrelogram in a standalone procedure, it may be included in the feature vectors of loss functions of other mosaic techniques and is thus of independent interest.

Secondly, we decompose the high-dimensional empirical autocorrelogram into a sparse dictionary of decaying sinusoids. By interpolating discrete signals, this procedure efficiently calculates both error and gradients with respect to time-scale parameters, enabling gradient-based optimization. The resulting technique is flexible and straightforward to parallelize on modern Single Instruction Multiple Data (SIMD) architectures such as Graphics Processing Units (GPUs).

We make our Python code[1] openly available for public use. We thereby aim to facilitate both the investigations of future researchers and the immediate application of these methods by musicians. Audio comparisons are made with benchmark mosaicing implementation, NiMFKS (Buch, Quinton, and Sturm 2017).

6.1 Prior work

Style transfer techniques, construed broadly, have a long history in audio signal processing research. Early work in this area begins with the channel vocoder (Dudley 1964), via various innovations to the modern repertoire of methods which includes recent advances such as neural style transfer methods (Engel et al. 2017; Grinstein et al. 2017; Verma and Smith 2018). In the context of the style transfer, mosaicing techniques form a sub field which fixes the choice of synthesis method to dictionary-based granular synthesis.

We are concerned specifically with the musical applications of style transfer. The archetypal task in this context is using the timbre of the 'style' signal to express the melodic 'content' of another. Concretely, if the target were a trumpet playing a melody, and the source a recording of a singing vocalist, the output should emulate the vocalist singing that melody. Hereafter we adopt the common convention that the style signal is the *source*, the content signal is the *target* and the synthesized hybrid is the *mosaic*.

In mosaicing synthesis, the task of choosing synthesis parameters to synthesise a mosaic with the desired properties is subject to ongoing research. Notable recent progress includes matrix factorization methods to decompose audio (Aarabi and Peeters 2018; Buch, Quinton, and Sturm 2017; Driedger and Pratzlich 2015), various improvements in spectral matrix factorization (Aarabi and Peeters 2018; Buch, Quinton, and Sturm 2017; Driedger and Pratzlich 2015) and optimization over feature space loss functions (Caetano and Rodet 2013; Coleman and Bonada 2008; Slaney, Covell, and Lassiter 1996). A restriction of the commonly available work is that few methods can conveniently handle time-scaling of audio, so that time-scale parameters must be ignored, or selected by exhaustive search over scaled

[1] https://github.com/danmackinlay/mosaicing_omp_ismir_2019/

copies of the source signal. One recent exception is Sound Retiler (Aarabi and Peeters 2018), which claims to handle time shifting via tensor decomposition, although this is not publicly available. It is in this area that we make our main contribution, by the application of autocorrelogram features in this task.

While the autocorrelogram itself is not new in audio synthesis (e.g. Slaney, Naar, and Lyon 1994), our application of it to the mosaicing problem is, to our knowledge, novel. The autocorrelogram-based analysis in combination with sparse coding induces a analytically differentiable expression for the time scale parameter, and it is this we use to solve the mosaic problems.

6.2 Problem description

6.2.1 Audio signals and notation

We work with audio signals, modelled as a Hilbert space $\mathcal{H}$ of real L_2 functions $f : \mathbb{R} \to \mathbb{R}$ mapping time to instantaneous signal pressure level (i.e., amplitude). Where the argument of the signal is clear, we abbreviate, writing for example, $t \mapsto f(at)$ as $f(at)$. We handle transforms on signals $f(\cdot)$ such as the autocorrelogram $\mathcal{A}$, and Fourier transform $\mathcal{F}$. Where not clear from context which argument of the signal with respect to which the transform is taken, we indicate it with a subscript to the transform. Thus $\mathcal{F}_t\{f(s,t)\}(\xi) \stackrel{\text{def}}{=} \int \mathrm{e}^{-2\pi i t\xi} f(s,t)\mathrm{d}t$. Where we specify a weight v for the inner product or norm, we write it as a subscript, i.e., $\langle f, g\rangle_v \stackrel{\text{def}}{=} \int_{\mathbb{R}} v(t)f(t)g(t)\mathrm{d}t$.

In practice we do not observe continuous audio signals, but discretely sampled observations of signals. Sampling in this context is meant in the signal processing sense, which means observing the value of some signal at certain co-ordinates, usually a lattice. The sense of 'sample' in the previous part of the theses, where it implies simulating a realization of some random variable is more usual in the stochastic simulation literature. Sampling fidelity is assumed, which is to say signals are band-limited to some suitably low cutoff period Ω, such that we may reconstruct them from the fixed-rate sampled version. We scale time so that the sample period is $t_{\max} = 1$ and $\Omega > 1/2$. The sampling process is a train of Dirac

impulses, and inner products with discrete signals are defined

$$\langle g, f \rangle_v \stackrel{\text{def}}{=} \sum_{t \in \mathbb{Z}} v(t) g(t) f(t). \tag{6.1}$$

We denote length-M vectors in bold, $\boldsymbol{x} = [x_1, x_2, \ldots, x_M]^\intercal$.

6.2.2 Mosaicing

Given a target signal f_0, we seek an approximant, the mosaic $\hat{f}_0$, as a sparse linear combination of scaled signals, called *codes*, from a source *dictionary* $\mathfrak{G} \stackrel{\text{def}}{=} \{g_1, \ldots, g_D\}$ subject to a maximum budget of J codes. In our earlier style transfer example, f_0 would be the recorded trumpet melody and $\mathfrak{G}$, recordings of the singing vocalist. For a fixed dictionary the mosaic is specified completely by the length-J parameter vectors $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}$ and written

$$\hat{f}_0(t; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}) = \sum_{j=1}^{J} \alpha_j g_{\gamma_j}(\rho_j t). \tag{6.2}$$

The problem requires selecting approximately optimal values for parameter vectors

$$\{\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}\} \simeq \underset{\{\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}\}}{\operatorname{argmin}} d\left(\hat{f}_0(t; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\rho}),\ f_0(t)\right), \tag{6.3}$$

where $\rho_j \in \mathbb{R}^+, \alpha_j \in \mathbb{R}, \gamma_j \in \{1, \ldots, D\}$ and $d : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}^+$ is a distance function quantifying the poorness of the approximation. In contrast to sparse coding for signal compression, $\hat{f}_0$ is an intentionally imperfect approximation of f_0, possessing qualities of both the source and target signals, i.e., *transferring* some elements of the "style".

Chapter 7

Autocorrelogram Mosaicing

We expand upon the autocorrelogram mosaicing method and the particular strengths it has in style transfer. This chapter is based on the work for the conference paper MacKinlay and Botev (2019) of which Dan MacKinlay is primary author. Some minor corrections and revisions for thesis style have been made.

7.1 Autocorrelation mosaicing method

The autocorrelogram mosaicing method has two stages.

1. In the pre-training stage, autocorrelogram features are computed from the source signals, and decomposed in a dictionary of decaying sinusoids.
2. In the inference stage, we search our dictionary of autocorrelogram decompositions for matches to the autocorrelogram of the target signal, and solve an inverse problem, synthesizing a corresponding mosaic from our result.

Both stages leverage convenient properties of autocorrelograms and sparse dictionary decompositions, which we now introduce.

7.2 Autocorrelograms

We now motivate the use of the autocorrelogram in our feature map. As with other style transfer methods we face the challenge that the sample values of a

time domain audio signal f are only indirectly indicative of how human listeners perceive it. For audio analysis one typically operates on a feature map $\mathcal{P}\{f\}$ which is in some sense closer to human perception of these signals. Specifically, we aim to find a feature map such that two signals are similar if some distance between their feature vectors is small, i.e., the similarity of f and $\hat{f}$ is high iff the distance $d_{\mathcal{P}}(f, \hat{f}) \stackrel{\text{def}}{=} \|\mathcal{P}\{f_0\} - \mathcal{P}\{\hat{f}\}\|$ is low, with some choice of norm $\|\cdot\|$. We would like $d_{\mathcal{P}}$ to approximate specifically *psychoacoustic* similarity — the smaller $d_{\mathcal{P}}(f, \hat{f})$, the greater the similarity between f and $\hat{f}$, from the perspective of a typical human listener. Ideally the feature map should also be of lower dimension than f, and $d_{\mathcal{F}}$ should be computationally efficient to manipulate.

True psychoacoustic similarity in this sense is not well-specified, so practical algorithms settle for feature maps compromising between convenience and psychoacoustic plausibility. Usually, feature maps are empirical PSDs (Caetano and Rodet 2013; Hoffman, Cook, and Blei 2009), or are derived from the PSD, as with the Mel-Frequency Cepstral Coefficient (MFCC) (Mermelstein and Chen 1976) or the Constant-Q transform (Brown 1991). These maps induce expensive mosaicing optimization problems (Coleman and Bonada 2008; Coleman, Maestre, and Bonada 2010). MFCCs for example, are suitable for low-dimensional indexing and search. They are hard to invert, in the sense of recovering a signal in the time domain given the low-dimensional feature representation, which makes them problematic for synthesis applications. A raw empirical PSD is easier to invert via Griffin-Lim iteration (Griffin and Lim 1984) or related methods. However the dimensionality of PSDs is not substantially lower than the original signal, and they are thus inconveniently large to store and index. One could ameliorate this difficulty if a computationally convenient feature map could be found which was well-behaved under operations of scaling and superposition, as in (6.2). In this case we could conduct more of the calculations in the low-dimensional feature space and still solve the inversion problem cheaply.

These desiderata suggest the autocorrelogram map,

$$\mathcal{A}\{f\} : \xi \mapsto \Big(\xi \mapsto \langle f(t), f(t-\xi)\rangle\Big). \tag{7.1}$$

This is the deterministic covariance between $f(t)$ and $f(t-\xi)$. The autocorrelogram

is an even function in ξ, so we work with one-sided autocorrelograms $\mathbb{R}^+ \to \mathbb{R}$. Autocorrelogram-like transforms are implicated in the neurological processing of harmonic audio by human listeners (Bidelman and Krishnan 2009; Cariani and Delgutte 1996; Langner 1992; Licklider 1951). For our purposes, the supposed neurological basis is a secondary consideration to the demonstrated empirical usefulness in psychoacoustic tasks, most notably in pitch-detection (Rabiner 1977; Slaney and Lyon 1990; Sondhi 1968). In this regard it resembles the cepstral analysis method (Bogert, Healy, and Tukey 1963), which also effectively identifies small numbers of periodic components by analysing a pointwise non-linear transformation of the power spectrogram. Unlike the cepstrum it is well-behaved under superposition.

Specifically, brief calculation (proved in Appendix C) shows the following useful properties.

$$\text{Scaling} \qquad \mathcal{A}\{cf\}(\xi) = c^2 \mathcal{A}\{f\}(\xi) \tag{7.2a}$$

$$\text{Dilation} \qquad \mathcal{A}\{f(rt)\}(\xi) = \frac{1}{r}\mathcal{A}\{f\}\left(\frac{\xi}{r}\right) \tag{7.2b}$$

$$\text{Randomized addition} \qquad \mathbb{E}\left[\mathcal{A}\{\mathsf{S}_1 f + \mathsf{S}_2 f'\}(\xi)\right] = \mathcal{A}\{f\}(\xi) + \mathcal{A}\{f'\}(\xi), \tag{7.2c}$$

Here f and f' are arbitrary signals, $c \in \mathbb{R}$ is an arbitrary constant and $\{\mathsf{S}_i\}$ are IID Rademacher variables, i.e., taking values in $\{+1, -1\}$ with equal probability.

We note another desirable feature of the autocorrelogram: it preserves symmetries and transformational invariances known to be important in perceptual audio analysis tasks. Finding such invariant features is an active area of research (e.g. Benetos, Cherla, and Weyde 2013; Lattner, Dorfler, and Arzt 2019; Luo, Agres, and Herremans 2019; Thickstun et al. 2017; Thickstun et al. 2018). Specifically, the autocorrelogram, like the power spectral density, is invariant to translation in time, since

$$\mathcal{A}\{f(t-s)\} = \mathcal{A}\{f(t-\xi-s)\} \tag{7.3}$$

since $\langle f(t-s), f(t-\xi-s)\rangle$. Human listeners also have translation-invariant hearing to a good approximation (e.g. a note of pitch $G\sharp$ under a millisecond delay is still

still has pitch $G\sharp$). Feature maps which can preserve such invariances are likely to be convenient for methods in feature space, e.g. learning operations which are indifferent to human-imperceptible changes in the signal.

We note two obstacles to the application of these formulae in the mosaicing problem. Firstly, (7.2b) is not well-defined for the discrete signals that comprise the usual subject matter of digital signal processing. We handle discrete signals by continuous interpolants, which turn out to be practically sufficient approximations. Secondly, the additive rule (7.2c) is valid only in expectation, via the contrivance of introducing Rademacher random variables. Solving for the deterministic case by accounting for phase cancellation is indeed possible, but considerably more involved, and constitutes an active area of research in its own right in the Overlap-Add (Driedger and Müller 2016; Verhelst and Roelands 1993) and phase retrieval (Jaganathan, Eldar, and Hassibi 2015; Shechtman et al. 2015) literatures. As the randomized solution also turns out in practice to be already sufficient for many tasks, we defer such extensions to future work.

In order to construct these interpolants efficiently, we decompose discrete autocorrelograms using a matching pursuit, which we now introduce.

7.3 Orthogonal matching pursuit

In orthogonal matching pursuit (OMP) (Davis, Mallat, and Zhang 1994; Pati, Rezaiifar, and Krishnaprasad 1993), given a target signal f_0 and a dictionary of code signals $\mathfrak{D} = \{g_\theta\}_{\theta\in\Theta}$, one finds a decomposition $\hat{f}_0 = \mathrm{OMP}_{\mathfrak{D},K}(f_0)$ of form

$$f_0 \simeq \mathrm{OMP}_{\mathfrak{D},K}(f_0) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{K} \mu_i g_{\theta_i}. \tag{7.4}$$

A solution is a parameter vector $[\theta_1, \ldots, \theta_K] \in \Theta^k$ and code weights $[\mu_1, \ldots, \mu_K] \in \mathbb{R}^K$ which nearly minimise $\|f_0 - \hat{f}_0\|$. We require that f_0 and all codes g_θ are L_2 integrable and not null, i.e., possessing positive norm, $\|g_\theta\| > 0$.

The OMP algorithm is as follows.

1. Initialization. Let the first residual be $r_0 \stackrel{\mathrm{def}}{=} f$. Set step counter $k \leftarrow 1$.
2. Find θ_k such that (possibly approximately)

$$\theta_k = \operatorname*{argmax}_{\theta} A(r_k, g_\theta) \tag{7.5}$$

where A is the *normalized code product*

$$A(r_k, g_\theta) \stackrel{\text{def}}{=} \frac{\langle r_{k-1}, g_\theta \rangle}{\|g_\theta\|}. \tag{7.6}$$

3. Solve the least sum of squares problem

$$[\mu_1^k, \ldots \mu_k^k] = \operatorname*{argmin}_{[\mu_1,\ldots,\mu_k]} \Big\| \sum_{1\le\ell\le k} \mu_\ell g_{\theta_\ell} - f_0 \Big\| \tag{7.7}$$

giving kth decomposition $\hat{f}^k = \sum_{1\le\ell\le k} \mu_\ell g_{\theta_\ell}$.

4. Update the residual $r_{k+1} = f_0 - \hat{f}^k$.

5. If $k = K$, stop, otherwise set $k \leftarrow k + 1$ and repeat from step (2).

We allow the components of θ to be either a) a discrete and finite, or b) a continuous parameter. For finitely enumerable components $\theta_{\text{finite}} \subseteq \theta$ we maximise the normalized code product in (7.5) by enumeration. For continuous components $\theta_{\text{cts}} \subseteq \theta$ we assume that we can choose θ_{cts} approximately by iterative optimization using the gradient $\nabla_{\theta_{\text{cts}}} A(r_k, g_\theta)$. As the objective may not attain a global maximum, we choose $I \geq 1$ different initial guesses, and select the best local optimum attained. A first order gradient ascent with a fixed number of steps performs well in our examples and moreover requires no branching instructions, as suits our goal of a SIMD-compatible algorithm.

7.4 Sparse approximate autocorrelograms

In the pre-training stage, we find autocorrelograms for each of the empirical source autocorrelogram codes in $\mathfrak{G}$, decomposing them into a dictionary of sparse OMP matches, $\mathfrak{M}$. It is this dictionary which we search for mosaic matches, using matches here to identify approximately matching codes in the original space $\mathfrak{G}$.

In this section we use ξ as the free argument for signals, and restrict $\xi > 0$. For the interpolant dictionary we use decaying sinusoids.

$$\mathfrak{S} \stackrel{\text{def}}{=} \{h(\xi; \omega, \tau, \phi) \stackrel{\text{def}}{=} \cos(\omega\xi + \phi)\mathrm{e}^{-\tau\xi} : \phi, \tau, \omega \in \mathbb{R}\}. \tag{7.8}$$

The dictionary choice must ultimately be justified by empirical performance, which we demonstrate in the final section of the chapter. It is notable that there are also *a priori* reasons for favouring this one for musical audio. Firstly, this basis decomposes an autocorrelogram into a global approximant, rather than a piecewise interpolant, as with, for example, polynomial splines. Evaluations of this interpolant are easy to parallelise without branching instructions, and therefore more natural for modern SIMD architectures.

Secondly, decaying sinusoid models are effective in compactly decomposing time-domain audio (Goodwin 1997), and the nature of the autocorrelogram suggests that they could be similarly useful and even more compact in decomposing autocorrelograms. The space of superpositions of decaying sinusoids is, by inspection, closed under the autocorrelogram transform, so it is just as plausible to represent autocorrelograms in such a decaying sinusoid dictionary. The question remains how compact such a representation is. Analytic expansion of the superposition of many decaying sinusoids is a lengthy exercise in elementary calculus. However, we suspect that the amplitude coefficient of most terms in such expansions are negligible. Recall the Wiener-Khintchine theorem, which states that, for signals of finite energy, assuming all these terms are well-defined,

$$\mathcal{F}_\xi\{\mathcal{A}\{f\}(\xi)\}(s) = |\mathcal{F}_t\{f(t)\}(s)|^2 \tag{7.9}$$

where $\mathcal{F}_\xi\{f(\xi)\}$ is the Fourier transform of signal $\xi \mapsto f(\xi)$. This tells us that the magnitude of sinusoidal components of the autocorrelogram are squared with respect to the magnitude of sinusoidal components of the PSD, and thus relatively sparser, in the sense that we can attain a close match whose squared error decays rapidly in the number of components. This indicates that for autocorrelograms of musical signals, which are well approximated by a superposition of sinusoidal signals, the autocorrelogram could often be approximated with comparable relative error by a yet smaller number of sinusoidal signals, as can be seen in Figure 7.1.

Moreover, we know that the envelope of musical audio spectral content decays eventually super-exponentially with frequency (Elowsson and Friberg 2017) and thus high frequency content of an autocorrelogram is proportionally even lower for musical content. This latter fact additionally implies that the autocorrelogram calculations might even be downsampled with little loss in information content, and some computational saving.

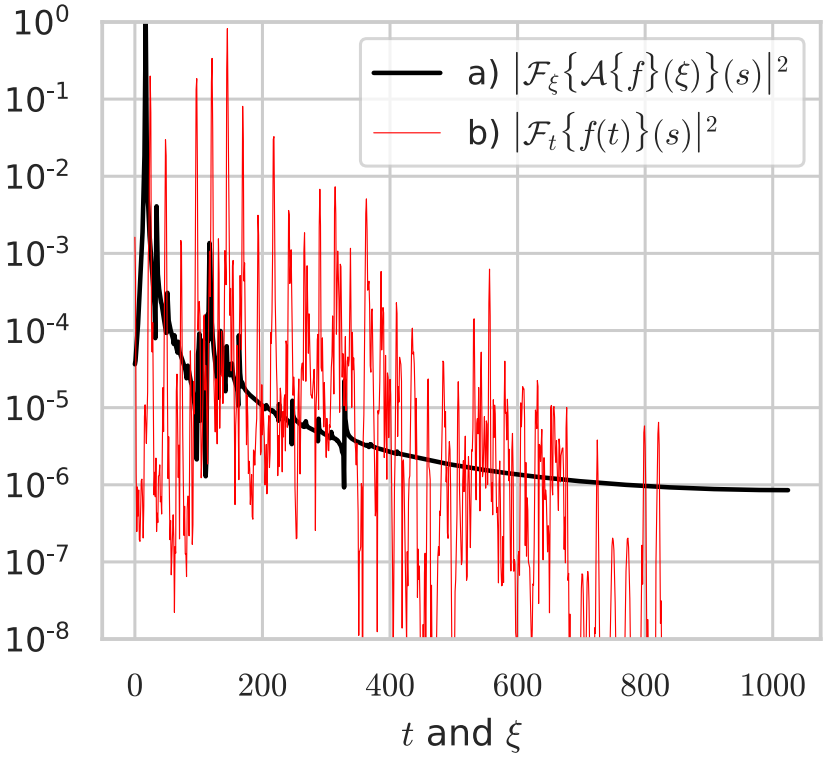


Figure 7.1: An example of a relatively simple form for (a) the PSD of the autocorrelogram versus (b) a complex PSD of the signal itself. The signal is a recording of a trumpet note onset. The scale of the vertical axis is arbitrary, and signals have been normalized for comparison. Sample period is $1/44100s$ and duration is 2048 time steps.

Implementing the decomposition is straightforward. For each code $g \in \mathfrak{G}$ we perform the following calculation:

First, we find the empirical autocorrelogram $\mathcal{A}\{g\}$ at L points $\xi = 0, 1, \ldots, (L-$

1) with (7.1).

Next, we decompose each $\hat{G} = \text{OMP}_{\mathfrak{S},C}(\mathcal{A}\{g\})$ over the decaying sinusoid dictionary, as defined in (7.8). There are many methods of fitting decaying sinusoids to time series (Barkhuijsen et al. 1985; Prony 1795; Serra and Smith 1990), but OMP is convenient in the current application (Goodwin 1997) as we may re-use the same algorithm in the reconstruction stage of this algorithm. Autocorrelograms of musical audio in our experiments are highly sparse with respect to this decaying sinusoid dictionary, typically achieving negligible residual error with a number of components $C \leq 4$. Analytic normalizations for such atoms are available in Section D.2.

We apply the OMP with product $\langle \cdot, \cdot \rangle_v$ weighted by $v(\xi) \stackrel{\text{def}}{=} \mathbb{I}\{[0, L)\}(\xi)/L$, returning parameters $\{\tau_i, \omega_i, \phi_i\}$ and code weights μ_i. We first find the normalized code product (7.6) in closed form. Substituting in (7.8) gives

$$A(r(\xi), h(\xi; \omega, \tau, \phi)\}) = \frac{\langle r_i(\xi), \cos(\omega\xi + \phi)\mathrm{e}^{\tau\xi} \rangle_v}{\| \cos(\omega\xi + \phi)\mathrm{e}^{\tau\xi} \|_v}. \tag{7.10}$$

The numerator is simply (6.1). Applying Euler identities gives the denominator

$$\| \cos(\omega\xi + \phi)\mathrm{e}^{-\tau\xi} \|_v^2 \tag{7.11}$$

$$= \frac{1}{2} \int_0^L (1 + \cos(2\omega\xi + 2\phi))\mathrm{e}^{-2\tau\xi} \mathrm{d}\xi \tag{7.12}$$

$$= \frac{\mathrm{e}^{-2\xi\tau}}{2} \frac{(\omega \sin(2\xi\omega + 2\phi) - \tau \cos(2\xi\omega + 2\phi))}{4\tau^2 + 4\omega^2} \Big|_{\xi=0}^{\xi=L} + \frac{1 - \mathrm{e}^{-2L\tau}}{4\tau}. \tag{7.13}$$

Combining (6.1) and (7.13) gives a closed form normalized code product (7.10), from which we can explicitly calculate gradients in τ, ω, ϕ as desired. Note that although the original signal is discrete, our decomposition is a continuous near-interpolant for it.

From these decompositions we construct the dictionary

$$\mathfrak{M} \stackrel{\text{def}}{=} \{\hat{G}_\gamma(\rho\xi) : \gamma \in (1, \ldots, D), \rho \in \mathbb{R}^+\}. \tag{7.14}$$

7.5 Synthesizing the mosaic

In the second stage, inference, we construct a mosaic $\hat{f}_0$ given a target f_0. Here we match the discrete autocorrelogram $F_0 \stackrel{\text{def}}{=} \mathcal{A}\{f_0\}$ by a second OMP decomposition $\hat{F}_0 \stackrel{\text{def}}{=} \text{OMP}_{\mathfrak{M},J}(F_0)$, into

$$\hat{F}_0(\xi) \stackrel{\text{def}}{=} \sum_{j=1}^{J} \kappa_j \hat{G}_{\gamma_j}(\rho_j \xi) \tag{7.15}$$

for index parameters $\{\gamma_i, \rho_i\}$ and weights κ_i. The OMP has already been introduced, but we pause to verify that it may be applied to this new context. Since each $\hat{G}_{\gamma_j}$ is a linear combination of decaying sinusoids (7.8), the normalizing denominator of the code product (7.6) is again a linear combination of decaying sinusoids, so its integral has a (lengthy) closed form as a linear combination of integrals (7.13), and we can find an explicit gradient $\nabla_\rho A(r_k, \rho)$. Thus we may find $\hat{F}_0$ as required.

Now we wish to construct $\hat{f}_0$ (6.2) such that

$$\mathbb{E}[\mathcal{A}\{\hat{f}_0\}] = \hat{F}_0. \tag{7.16}$$

Choosing $\hat{f}_0 \stackrel{\text{def}}{=} \sum_j \mathsf{S}_j \alpha_j g_{\gamma_j}(\rho_j t)$ by matching pursuit, simulating S_j independent Rademacher variates, and applying (7.2a) (7.2b) and (7.2c) to (6.2), we find

$$\begin{aligned} \mathbb{E}[\mathcal{A}\{\hat{f}_0\}] &= \mathbb{E}\left[\mathcal{A}\left\{\sum_j \mathsf{S}_j \alpha_j g_{\gamma_j}(\rho_j t)\right\}(\xi)\right] \\ &= \sum_j \frac{\alpha_j^2}{\rho_j} \mathcal{A}\left\{g_{\gamma_j}(t)\right\}(\rho_j \xi) \\ &\simeq \sum_j \frac{\alpha_j^2}{\rho_j} \hat{G}_{\gamma_j}(\rho_j \xi). \end{aligned} \tag{7.17}$$

By inspection,

$$\alpha_j = \mathsf{S}_j \sqrt{|\rho_j||\kappa_j|} \tag{7.18}$$

satisfies (7.16). We resample the original discrete dictionary codes to target time scale ρ_i by band-limited sinc interpolation (Smith 2018). Finally, we substitute

the resulting α_j into (6.2) and superpose grains to realise the desired mosaic.

7.6 Localized matching

So far we have discussed entire signals, implicitly assuming them to be brief. The autocorrelogram, taken globally over a long signal such as an entire musical piece, no longer estimates the local, stylistic characteristics. Just as one adapts the discrete Fourier transform for long signals into the Short-Time Fourier Transform (STFT) (Blackman and Tukey 1959), so do we adapt the autocorrelogram mosaic method, applying it locally. A simple localization is to slice signals into short frames of fixed duration M, which are called *grains* by convention. As in the STFT, we multiply each frame point-wise with real window function w, supported on $[0, M]$ with $\|w\| = 1$. Hereafter, we assume a *sine window*, $w(t) \stackrel{\text{def}}{=} 2\sin(\pi t/M)\mathbb{I}[0, M]/M$. We fix hop length $H < M$. Next, we localize $\mathfrak{G}$ into a new dictionary whose codes are precisely these time-shifted grains (disallowing zero-energy grains).

$$\mathfrak{G}^{w,H} \stackrel{\text{def}}{=} \{w(t)g(t-\phi) : g \in \mathfrak{G}, \phi/H \in \mathbb{Z}, \|g'\| > 0\}. \tag{7.19}$$

In musical material a localized dictionary tends to high redundancy and marginal return on search effort decreases. Rather than proceeding exhaustively, we keep the search tractable by searching a pseudorandom subset of fixed size, where the size of this pseudorandom subset is a user selectable parameter.

In the synthesis stage, we localize the target signal, $f_0^w(t;\phi) \stackrel{\text{def}}{=} w(t)f_0(t-\phi)$, constructing a local mosaic $\hat{f}_0^w(t;\phi)$ from $\mathfrak{G}^{w,H}$ for $\phi \in \{0, H, 2H, \dots\}$. Finally, we superpose the local mosaics into a global one,

$$\hat{f}_0(t) = \sum_{\ell \in \mathbb{Z}} \hat{f}_0^w(t + H\ell; H\ell). \tag{7.20}$$

7.7 Experiments

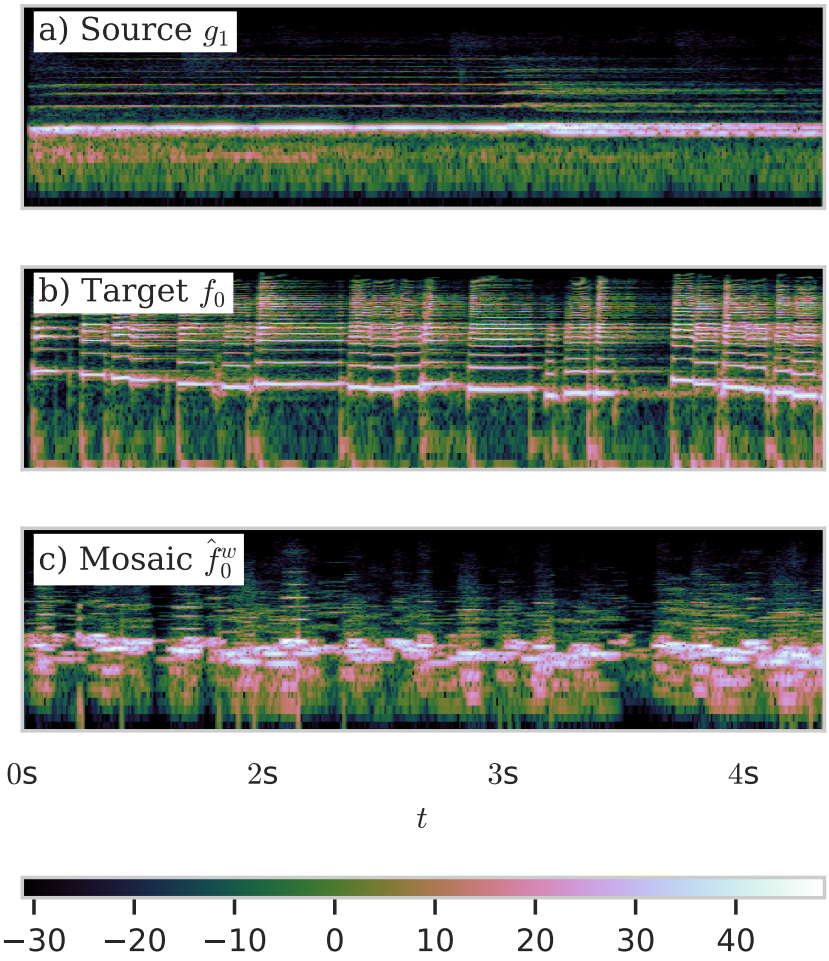


Figure 7.2: Power spectral density of signals a) source vocal recording b) target trumpet recording and c) resulting mosaic. Frequency increases up vertical axis, intensity in dB with arbitrary normalization.

As an initial example we transfer style with target signal f_0 as a trumpet solo[1] and source signal $\{G\}$ as a vocal recording.[2] Audio is sampled with a period of 1/44100s. We fix parameters, taking grain size $M = 8192$, hop length $H = M/2$,

[1] credit Mihai Sorohan
[2] credit Emm Collins

correlogram length $L = 1024$, number of sinusoidal components $C = 4$, matching grains $J = 1$, and number of initial guesses $I = 12$. Optimization routines are left at system defaults. Examining the spectrogram Figure 7.2 illustrates phenomena compatible with our claims: In the mosaic we observe local features of the source with the larger structure of the target, to wit, the pitch contours of the trumpet solo with a spectral distribution approximating a human voice.

We next apply the algorithm across a small corpus and compare our results against the mosaicing algorithm NiMFKS (Buch, Quinton, and Sturm 2017). NiMFKS is a useful benchmark for mosaicing synthesis, incorporating many different user-selectable loss functions and decomposition methods from elsewhere in the literature, and possessing openly available code.[3] Their method generalizes classical mosaicing by using a non negative PSD factorization to further decompose grains into a sparse product of activations and responses. Unlike our method it does not infer optimal time scaling of audio.

Performance evaluation of mosaicing methods is subjective. In the following, we nevertheless attempt to describe the behaviors of the two algorithms as objectively as we are able. In order to challenge the NiMFKS model, our corpus samples are tuned to a variety of different root notes, scales and audio ranges, including Indonesian, Western and centreless[4] tunings. Style transfer is applied to every pairing of samples. Parameters are left at default values in NiMFKS. Parameters for the autocorrelogram mosaic are specified when they occur. These may be heard in the online supplemental material. Subjectively, neither method seems to produce naturalistic outputs for all pairs of source and target audio. NiMFKS seems ascendant where the source audio is polyphonic and the factorization succeeds at decomposing different notes where our method cannot. On the other hand, where the target tuning is not spanned by the source, the sparse autocorrelogram method is able to produce smoother and better related mosaics by transposing source grains to match the target. Occasionally the sparse autocorrelogram mosaics sound rough during rapid articulations. The method could possibly be improved in these cases by adaptive selection of grain size, or tuning of the free hyperparameters in the model, or extension with non-randomized re-

[3] https://code.soundsoftware.ac.uk/projects/nimfks

[4] i.e. recordings without a standard “root” notes

construction methods. Even in these cases, however, simultaneous playback of the target and the mosaic reveals that we maintain harmonic relationships with the target audio. As such, even this imperfect reconstruction can be regarded as an exotic musical effect. In summary, even at this early stage, our method succeeds in extending mosaic methods to previously intractable tasks, and produces musically interesting output. Work remains to be done in improving performance and integrating with other methods.

7.8 Conclusions: Autocorrelogram mosaicing

This trick extends the reach of mosaicing methods by demonstrating advantages of the autocorrelogram feature map for solving problems in audio analysis. By combining autocorrelogram feature maps and interpolating matching pursuit in particular, we have extended the library of methods of audio mosaicing style transfer. Our method in isolation produces interesting results on the sample data with little tuning. Work remains to be done in analysing the robustness and generality of the method, and selecting optimal trade-off of cost and quality of different style transfer tasks under different choices of user parameters. More work also remains to be done in integrating this method with existing ones. The array of loss functions supported by, for example, NiMFKS, could be augmented to include autocorrelogram features, and the autocorrelogram approach can be applied to spectrally decomposed signals, which are still audio signals. However, the ease with which we produce good results suggests that further extensions and refinements are worthy of pursuit. Should that source code become available,i t would be instructive to compare against mosaicing method Music Retiler (Aarabi and Peeters 2018) which claims to handle time scaling of audio via a different method.

This method has been phrased in terms of deterministic signal processing, but the discussion of autocorrelation functions evokes models of stochastic processes (Yaglom 1987), and indeed is reminiscent of the work in Gaussian process regression (Rasmussen and Williams 2006). This method could likely benefit from a fully Bayesian treatment as a stochastic signal processing method in a probabilistic function space. Lately fully probabilistic spectral analyses have made some progress in related problems of audio analysis in the context of probabilistic spec-

tral analysis (Alvarado, Alvarez, and Stowell 2019; Liutkus et al. 2014; Wilkinson et al. 2019). Incorporating rich priors and a principled probabilistic model of the signal is thus a provocative potential avenue for further research.

Appendix C

Properties of autocorrelograms

Consider an L_2 signal $f : \mathbb{R} \to \mathbb{R}$. We overload notation and write it with free argument t, so that $f(rt - \xi)$, for example, refers to the signal $t \mapsto f(rt - \xi)$.

The autocorrelogram transform $\mathcal{A} : L_2(\mathbb{R}) \to L_2(\mathbb{R})$ maps signals to signals. Specifically, $\mathcal{A}\{f\}$ is a signal $\mathbb{R} \to \mathbb{R}$ such that

$$\mathcal{A}\{f\} \stackrel{\text{def}}{=} \xi \mapsto \langle f(t), f(t-\xi)\rangle \tag{C.1}$$

This is the (deterministic) covariance between $f(t)$ and $f(t-\xi)$. We list the properties of this transform.

Proposition C.1 (Multiplication by a constant)**.** Consider a constant $c \in \mathbb{R}$.

$$\begin{aligned}
\mathcal{A}\{cf\}(\xi) &= \langle cf(t), cf(t-\xi)\rangle && \text{(C.2)}\\
&= c^2\langle f(t), f(t-\xi)\rangle && \text{(C.3)}\\
&= c^2\mathcal{A}\{f\}(\xi). && \text{(C.4)}
\end{aligned}$$

Proposition C.2 (Time scaling)**.**

$$\mathcal{A}\{f(rt)\}(\xi) = \langle f(rt), f(rt-\xi)\rangle \tag{C.5}$$
$$= \int f(rt)f(rt-\xi)\mathrm{d}t \tag{C.6}$$
$$= \frac{1}{r}\int f(t)f(t-\frac{\xi}{r})\mathrm{d}t \tag{C.7}$$
$$= \frac{1}{r}\mathcal{A}\{f\}\left(\frac{\xi}{r}\right). \tag{C.8}$$

Proposition C.3 (Addition)**.**

$$\mathcal{A}\{f+f'\}(\xi) = \langle f(t)+f'(t), f(t-\xi)+f'(t-\xi)\rangle \tag{C.9}$$
$$= \langle f(t), f(t-\xi)\rangle + \langle f(t), f'(t-\xi)\rangle + \langle f'(t), f(t-\xi)\rangle + \langle f'(t), f'(t-\xi)\rangle \tag{C.10}$$
$$= \mathcal{A}\{f\}(\xi) + \langle f'(t), f(t-\xi)\rangle + \langle f(t), f'(t-\xi)\rangle + \mathcal{A}\{f'\}(\xi). \tag{C.11}$$
$$= \mathcal{A}\{f\}(\xi) + \langle f'(t), f(t-\xi)\rangle + \langle f(t+\xi), f'(t)\rangle + \mathcal{A}\{f'\}(\xi). \tag{C.12}$$
$$= \mathcal{A}\{f\}(\xi) + \langle f'(t), f(t-\xi)\rangle + \langle f'(t), f(t+\xi)\rangle + \mathcal{A}\{f'\}(\xi). \tag{C.13}$$

We can say little about the term $\langle f'(t), f(t-\xi)\rangle + \langle f'(t), f(t+\xi)\rangle$ without more information about the signals in question. However, we can solve a randomized version. Suppose S_i, $i \in \mathbb{N}$ are IID Rademacher variables, i.e., that they assume a value in $\{+1, -1\}$ with equal probability. Then, we have the following randomized version.

Proposition C.4 (Addition)**.**

$$\begin{aligned}
\mathbb{E}[\mathcal{A}\{S_1 f + S_2 f'\}(\xi) = \mathbb{E}[&\mathcal{A}\{S_1 f\}(\xi) + \\
&\langle S_2 f'(t), S_1 f(t-\xi)\rangle \\
&+ \langle S_2 f'(t), S_1 f(t+\xi)\rangle + \mathcal{A}\{S_2 f'\}(\xi)] && \text{(C.14)}\\
= \mathbb{E}[&\mathcal{A}\{S_1 f\}(\xi)] + \mathbb{E}\langle S_2 f'(t), S_1 f(t-\xi)\rangle \\
&+ \mathbb{E}\langle S_2 f'(t), S_1 f(t+\xi)\rangle + \mathbb{E}[\mathcal{A}\{S_2 f'\}(\xi)] && \text{(C.15)}\\
= \mathcal{A}\{f\}(\xi) &+ \mathbb{E}[S_1 S_2]\langle f'(t), f(t-\xi)\rangle \\
&+ \mathbb{E}[S_1 S_2]\langle f'(t), f(t+\xi)\rangle + \mathcal{A}\{f'\}(\xi) && \text{(C.16)}\\
= \mathcal{A}\{f\}(\xi) &+ \mathcal{A}\{f'\}(\xi). && \text{(C.17)}
\end{aligned}$$

The sum of expectations, that is, is not guaranteed to have a simple deterministic form, but the expectation of the randomized version is. This restriction is shared with many mosaicing methods based on DFT power spectrograms, which discard or randomise phase information; e.g. this is entailed by the additive-power assumption of (Hoffman, Cook, and Blei 2009).

Appendix D

Decaying sinusoidal basis

Using Euler identities we find the antiderivative

$$\int^{L} \cos(\omega\xi)\exp(-\tau\xi)\mathrm{d}\xi = \frac{\mathrm{e}^{-L\tau}(\omega\sin(L\omega+\phi)-\tau\cos(L\omega+\phi))}{\tau^2+\omega^2} \tag{D.1}$$

and thus

$$\begin{aligned}\int_0^{L} \cos(\omega\xi)\exp(-\tau\xi)\mathrm{d}\xi &= \frac{1}{\tau^2+\omega^2}\mathrm{e}^{-\xi\tau}(\omega\sin(\xi\omega+\phi)-\tau\cos(\xi\omega+\phi))\Big|_{\xi=0}^{\xi=L} &\text{(D.2)}\\ &= \frac{1}{\tau^2+\omega^2}\Big(\mathrm{e}^{-L\tau}(\omega\sin(L\omega+\phi)-\tau\cos(L\omega+\phi)) \\ &\quad -\omega\sin\phi+\tau\cos\phi)\Big). &\text{(D.3)}\end{aligned}$$

D.1 Inner products of decaying sinusoidal atoms

We find norms for the atoms by mechanically calculation.

$$
\begin{aligned}
&\langle \cos(\omega\xi+\phi)\exp -\tau\xi, \cos(\omega'\xi'+\phi')\exp -\tau'\xi\rangle \\
&= \frac{1}{2}\int v(\xi)\Big(\cos(\omega\xi+\phi-\omega'\xi-\phi') \\
&\qquad + \cos(\omega\xi+\phi+\omega'\xi'+\phi')\Big)\exp(-(\tau'+\tau)\xi)\mathrm{d}\xi \qquad \text{(D.4)} \\
&= \frac{1}{2}\int v(\xi)\Big(\cos((\omega-\omega)\xi+\phi-\phi') \\
&\qquad + \cos((\omega+\omega')\xi+\phi+\phi')\Big)\exp(-(\tau'+\tau)\xi)\mathrm{d}\xi \qquad \text{(D.5)} \\
&= \frac{1}{2}\int v(\xi)\cos((\omega-\omega')\xi+\phi-\phi')\exp(-(\tau'+\tau)\xi)\mathrm{d}\xi \\
&\qquad + \frac{1}{2}\int v(\xi)\cos((\omega+\omega')\xi+\phi+\phi')\exp(-(\tau'+\tau)\xi)\mathrm{d}\xi \qquad \text{(D.6)} \\
&= \frac{1}{2}\int v(\xi)\cos(\omega_-\xi+\phi_-)\exp(-\tau_+\xi)\mathrm{d}\xi \\
&\qquad + \frac{1}{2}\int v(\xi)\cos(\omega_+\xi+\phi_+)\exp(-\tau_+\xi)\mathrm{d}\xi. \qquad \text{(D.7)}
\end{aligned}
$$

Here we have defined the abbreviations $\phi_+ \stackrel{\text{def}}{=} \phi+\phi'$, $\phi_- \stackrel{\text{def}}{=} \phi-\phi'$, $\omega_+ \stackrel{\text{def}}{=} \omega+\omega'$, $\omega_- \stackrel{\text{def}}{=} \omega-\omega'$ and $\tau_+ \stackrel{\text{def}}{=} \tau-\tau'$.

If we choose "top hat" weight $v = \mathbb{I}[0, L]$, we may expand this using (D.3)

$$\begin{aligned}
&\langle \cos(\omega\xi + \phi) \exp -\tau\xi, \cos(\omega'\xi' + \phi') \exp -\tau'\xi \rangle_v \\
&= \frac{1}{2} \int_0^L \cos(\omega_-\xi + \phi_-) \exp(-\tau_+\xi) \mathrm{d}\xi \\
&\quad + \frac{1}{2} \int_0^L \cos(\omega_+\xi + \phi_+) \exp(-\tau_+\xi) \mathrm{d}\xi \qquad \text{(D.8)} \\
&= \frac{1}{2} \frac{\mathrm{e}^{-\xi\tau_+} \left(\omega_- \sin(\xi\omega_- + \phi_-) - \tau_+ \cos(\xi\omega_- + \phi_-)\right)}{\tau_+^2 + \omega_-^2} \Bigg|_{\xi=0}^{\xi=L} \\
&\quad + \frac{1}{2} \frac{\mathrm{e}^{-\xi\tau_+} \left(\omega_+ \sin(\xi\omega_+ + \phi_+) - \tau_+ \cos(\xi\omega_+ + \phi_+)\right)}{\tau_+^2 + \omega_+^2} \Bigg|_{\xi=0}^{\xi=L} \qquad \text{(D.9)} \\
&= \frac{1}{2(\tau_+^2 + \omega_-^2)} \Big(\mathrm{e}^{-L\tau_+} \left(\omega_- \sin(L\omega_- + \phi_-) - \tau_+ \cos(L\omega_- + \phi_-)\right) - \omega_- \sin\phi_- + \tau_+ \cos\phi_- \Big) \\
&\quad + \frac{1}{2(\tau_+^2 + \omega_+^2)} \Big(\mathrm{e}^{-L\tau_+} \left(\omega_+ \sin(L\omega_+ + \phi_+) - \tau_+ \cos(L\omega_+ + \phi_+)\right) - \omega_+ \sin\phi_+ + \tau_+ \cos\phi_+ \Big). \\
&\qquad \text{(D.10)}
\end{aligned}$$

D.2 Normalizing decaying sinusoidal atoms

In matching pursuit we need to calculate inner products with normalized codes (7.6). This means evaluating the denominator norm

$$\begin{aligned}
\| \cos(\omega\xi + \phi) \exp \tau\xi \|_v^2 &= \int v(\xi) \left(\cos(\omega\xi + \phi) \exp \tau\xi\right)^2 \mathrm{d}\xi \qquad \text{(D.11)} \\
&= \int v(\xi) \cos^2(\omega\xi + \phi) \exp(2\tau\xi) \mathrm{d}\xi \qquad \text{(D.12)} \\
&= \frac{1}{2} \int v(\xi)(1 + \cos(2\omega\xi + 2\phi)) \exp(-2\tau\xi) \mathrm{d}\xi. \qquad \text{(D.13)}
\end{aligned}$$

If we choose, for example, a top hat weight $v = \mathbb{I}[0, L]$ we can specialise (D.13)

to give (7.10),

$$\| \cos(\omega\xi + \phi) \exp -\tau\xi \|_v^2 = \int_0^L \left(\cos(\omega\xi + \phi) \exp(-\tau\xi)\right)^2 \mathrm{d}\xi \tag{D.14}$$

$$= \frac{1}{2} \int_0^L (1 + \cos(2\omega\xi + 2\phi)) \exp(-2\tau\xi) \mathrm{d}\xi \tag{D.15}$$

$$= \frac{1}{2} \int_0^L \mathrm{e}^{-2\xi\tau} \cos(2\xi\omega + 2\phi) + \mathrm{e}^{-2\xi\tau} \mathrm{d}\xi \tag{D.16}$$

$$= \frac{1}{2} \int_0^L \mathrm{e}^{-2\xi\tau} \cos(2\xi\omega + 2\phi) \mathrm{d}\xi + \frac{1}{2} \int_0^L \mathrm{e}^{-2\xi\tau} \mathrm{d}\xi \tag{D.17}$$

$$= \frac{\mathrm{e}^{-2\xi\tau}}{2} \frac{(\omega \sin(2\xi\omega + 2\phi) - \tau \cos(2\xi\omega + 2\phi))}{4\tau^2 + 4\omega^2} \Bigg|_{\xi=0}^{\xi=L} + \frac{1 - \mathrm{e}^{-2L\tau}}{4\tau}. \tag{D.18}$$

We deduce closed form solutions for other weight functions based on, for example, trigonometric functions, although these become tedious to write out in full.

Appendix E

Normalizing decaying sinusoidal molecules

We consider a signal F which is a molecule comprising superposed decaying sinusoid atoms, i.e. $F : \xi \mapsto \sum_{k=1}^{K} \alpha_k \cos(\omega_k \xi + \phi_k) \exp \tau_k \xi$. To normalize this molecule we use linearity of inner products,

$$\langle F, F \rangle = \left\langle \sum_k \alpha_k \cos(\omega_k \xi + \phi_k) \exp \tau_k \xi, \sum_k \alpha_k \cos(\omega_k \xi + \phi) \exp \tau_k \xi \right\rangle \tag{E.1}$$

$$= \sum_{j,k} \alpha_j \alpha_k \left\langle \cos(\omega_j \xi + \phi_j) \exp \tau_j \xi, \cos(\omega_k \xi + \phi_k) \exp \tau_k \xi \right\rangle \tag{E.2}$$

$$= 2 \sum_{k=1}^{K} \sum_{j<k} \alpha_j \alpha_k \left\langle \cos(\omega_j \xi + \phi_j) \exp \tau_j \xi, \cos(\omega_k \xi + \phi_k) \exp \tau_k \xi \right\rangle + \sum_{k=1}^{K} \alpha_k^2 \| \cos(\omega_k \xi_k + \phi_k) \exp -\tau_k \xi_k \|^2. \tag{E.3}$$

Using, for example, the top hat weight function $v = \mathbb{I}[0, L]$ we can apply (D.18) and (7.6) to find a (lengthy) closed-form expression for this normalizing term.

Chapter 8

Conclusion

The two parts of this thesis have diverged greatly from one another in the course of development, starting from a common origin in models for time-series data. In the first we expounded a method for solving certain rare event estimation problems with a set of convenient assumptions which lead to nearly-automatic design and approximate optimization of the sampling parameters. In the second we used autocorrelation features and a stochastic simulation method to solve approximation problems in audio analysis. While we hold an abiding belief in the unity of stochastic approximation methods for time series analysis, a synthesis of these two parts is beyond the reach of a humble graduate thesis. For now they stand as separate endeavours, united only by a few shared tools in probability theory. In both these domains we demonstrate the usefulness of randomized approximations, and as such we map tributaries of the greater river of stochastic distributional learning. Somewhere these tributaries meet in a great confluence of Monte Carlo methods. Now we set sail, downstream.

References

Aalen, Odd O., Ørnulf Borgan, and S. Gjessing (2008). *Survival and Event History Analysis: A Process Point of View*. Statistics for Biology and Health. New York, NY: Springer. 539 pp. ISBN: 978-0-387-20287-7 (cit. on p. 89).

Aarabi, Hadrien Foroughmand and Geoffroy Peeters (2018). "Music Retiler: Using NMF2D Source Separation for Audio Mosaicing". In: *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion* (Wrexham, United Kingdom). AM'18. New York, NY, USA: ACM, 27:1–27:7. ISBN: 978-1-4503-6609-0. DOI: 10.1145/3243274.3243299 (cit. on pp. 125, 127, 128, 143).

Adlakha, V. G. and V. G. Kulkarni (1989). "A Classified Bibliography Of Research On Stochastic Pert Networks: 1966-1987". In: *INFOR: Information Systems and Operational Research* 27.3 (3), pp. 272–296. DOI: 10.1080/03155986.1989.11732098 (cit. on p. 70).

Alvarado, Pablo A., Mauricio A. Alvarez, and Dan Stowell (2019). "Sparse Gaussian Process Audio Source Separation Using Spectrum Priors in the Time-Domain". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 995–999. DOI: 10.1109/ICASSP.2019.8683287 (cit. on p. 144).

Applebaum, David (2009). *Lévy Processes and Stochastic Calculus*. 2nd ed. Cambridge Studies in Advanced Mathematics 116. Cambridge ; New York: Cambridge University Press. 460 pp. ISBN: 978-0-521-73865-1 (cit. on pp. 116, 118).

Asmussen, Søren and Peter W. Glynn (2007). *Stochastic Simulation: Algorithms and Analysis*. 2007 edition. New York: Springer. 476 pp. ISBN: 978-0-387-30679-7 (cit. on pp. 12, 24, 26, 28, 39, 116, 118).

Asmussen, Søren and Reuven Y. Rubinstein (1995). "Steady State Rare Events Simulation in Queueing Models and Its Complexity Properties". In: *In Advances in Queueing*. CRC Press, pp. 429–461 (cit. on p. 15).

Balkema, A. A. and L. de Haan (1974). "Residual Life Time at Great Age". In: *The Annals of Probability* 2.5 (5), pp. 792–804. DOI: 10.1214/aop/1176996548 (cit. on p. 83).

Barkhuijsen, H. et al. (1985). "Retrieval of Frequencies, Amplitudes, Damping Factors, and Phases from Time-Domain Signals Using a Linear Least-Squares Procedure". In: *Journal of Magnetic Resonance (1969)* 61.3 (3), pp. 465–481. DOI: 10.1016/0022-2364(85)90187-8 (cit. on p. 138).

Bashir, Muhammad Salman and Mohamed-Slim Alouini (2020). "Signal Acquisition With Photon-Counting Detector Arrays in Free-Space Optical Communications". In: *IEEE Transactions on Wireless Communications* 19.4 (4), pp. 2181–2195. DOI: 10.1109/TWC.2019.2962670 (cit. on p. 63).

Ben Rached, N. et al. (2016). "Unified Importance Sampling Schemes for Efficient Simulation of Outage Capacity over Generalized Fading Channels". In: *IEEE Journal of Selected Topics in Signal Processing* 10.2 (2), pp. 376–388. DOI: 10.1109/JSTSP.2015.2500201 (cit. on pp. 62, 63).

— (2017). "On the Efficient Simulation of Outage Probability in a Log-Normal Fading Environment". In: *IEEE Transactions on Communications* 65.6 (6), pp. 2583–2593. DOI: 10.1109/TCOMM.2017.2669979 (cit. on pp. 62, 67).

Ben Rached, Nadhir et al. (2018a). "Importance Sampling Estimator of Outage Probability under Generalized Selection Combining Model". In: *Proc of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 5. DOI: 10.1109/ICASSP.2018.8462177. URL: https://repository.kaust.edu.sa/bitstream/handle/10754/630800/Conference_paper.pdf?sequence=1 (cit. on p. 64).

— (2018b). "On the Sum of Order Statistics and Applications to Wireless Communication Systems Performances". In: *IEEE Transactions on Wireless Communications* 17.11 (11), pp. 7801–7813. DOI: 10.1109/TWC.2018.2871201. URL: http://arxiv.org/abs/1711.04280 (visited on 02/10/2019) (cit. on pp. 62–65).

Ben Rached, Nadhir et al. (2020). "A Universal Splitting Estimator for the Performance Evaluation of Wireless Communications Systems". In: *IEEE Transactions on Wireless Communications*. DOI: 10.1109/TWC.2020.2982649. URL: https://arxiv.org/abs/1908.10616v1 (cit. on pp. xi, 2, 7, 57, 93).

Benetos, Emmanouil, Srikanth Cherla, and Tillman Weyde (2013). "An Efficient Shift-Invariant Model for Polyphonic Music Transcription". In: *6th International Workshop on Machine Learning and Music*. 6th International Workshop on Machine Learning and Music. Prague, Czech Republic, p. 4 (cit. on p. 133).

Bertoin, Jean (1996). *Lévy Processes*. Cambridge Tracts in Mathematics 121. Cambridge ; New York: Cambridge University Press. 265 pp. ISBN: 978-0-521-56243-0 (cit. on pp. 50, 117).

Bhatti, Sajjad Haider et al. (2018). "Efficient Estimation of Pareto Model: Some Modified Percentile Estimators". In: *PLOS ONE* 13.5 (5), e0196456. DOI: 10.1371/journal.pone.0196456 (cit. on p. 112).

Bidelman, Gavin M. and Ananthanarayan Krishnan (2009). "Neural Correlates of Consonance, Dissonance, and the Hierarchy of Musical Pitch in the Human Brainstem". In: *Journal of Neuroscience* 29.42 (42), pp. 13165–13171. DOI: 10.1523/JNEUROSCI.3900-09.2009. pmid: 19846704 (cit. on p. 133).

Bithas, Petros S., Nikos C. Sagias, and Takis P. Mathiopoulos (2007). "GSC Diversity Receivers over Generalized-Gamma Fading Channels". In: *IEEE Communications Letters* 11.12 (12), pp. 964–966 (cit. on p. 63).

Blackman, R. B. and J. W. Tukey (1959). *The Measurement of Power Spectra from the Point of View of Communications Engineering*. New York: Dover Publications. 190 pp. ISBN: 978-0-486-60507-4 (cit. on p. 140).

Bogert, B P, M J R Healy, and J W Tukey (1963). "The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum and Saphe Cracking". In: Symposium on Time Series Analysis, pp. 209–243 (cit. on p. 133).

Botev, Zdravko and Pierre L'Ecuyer (2017). "Simulation from the Normal Distribution Truncated to an Interval in the Tail". In: *Proceedings of the 10th EAI International Conference on Performance Evaluation Methodologies and Tools*. VALUETOOLS'16. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineer-

ing), pp. 23–29. ISBN: 978-1-63190-141-6. DOI: 10.4108/eai.25-10-2016.2266879 (cit. on p. 74).

Botev, Zdravko and Ad Ridder (2017). "Variance Reduction". In: *Wiley StatsRef: Statistics Reference Online*. American Cancer Society, pp. 1–6. ISBN: 978-1-118-44511-2. DOI: 10.1002/9781118445112.stat07975. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat07975 (visited on 06/08/2020) (cit. on p. 21).

Botev, Zdravko I, Pierre L'Ecuyer, and Bruno Tuffin (2018). *Reliability Estimation for Networks with Minimal Flow Demand and Random Link Capacities*. arXiv: 1805.03326 (cit. on pp. 9, 70).

Botev, Zdravko I. and Dirk P. Kroese (2008). "An Efficient Algorithm for Rare-Event Probability Estimation, Combinatorial Optimization, and Counting". In: *Methodology and Computing in Applied Probability* 10.4 (4), pp. 471–505. DOI: 10.1007/s11009-008-9073-7. URL: http://ie.technion.ac.il/CE/files/adam.pdf (cit. on p. 40).

— (2012). "Efficient Monte Carlo Simulation via the Generalized Splitting Method". In: *Statistics and Computing* 22.1 (1), pp. 1–16. DOI: 10.1007/s11222-010-9201-4. URL: https://people.smp.uq.edu.au/DirkKroese/ps/generalized_splitting.pdf (cit. on p. 23).

Botev, Zdravko I., Pierre L'Ecuyer, and Bruno Tuffin (2012). "Dependent Failures in Highly Reliable Static Networks". In: *Proceedings of the 2012 Winter Simulation Conference (WSC)*. 2012 Winter Simulation Conference - (WSC 2012). Berlin, Germany: IEEE, pp. 1–12. DOI: 10.1109/WSC.2012.6465033 (cit. on pp. 34, 35, 39).

Botev, Zdravko I. and Pierre L'Ecuyer (2020). "Sampling Conditionally on a Rare Event via Generalized Splitting". In: *INFORMS Journal on Computing*. DOI: 10.1287/ijoc.2019.0936. arXiv: 1909.03566. URL: http://arxiv.org/abs/1909.03566 (visited on 06/01/2020) (cit. on pp. 23, 38).

Botev, Zdravko I., Robert Salomone, and Daniel Mackinlay (2019). "Fast and Accurate Computation of the Distribution of Sums of Dependent Log-Normals". In: *Annals of Operations Research*. DOI: 10.1007/s10479-019-03161-x. URL: https://rdcu.be/bl3o9 (visited on 02/11/2019) (cit. on pp. 62, 67, 74).

Botev, Zdravko I. et al. (2012). "Static Network Reliability Estimation via Generalized Splitting". In: *INFORMS Journal on Computing* 25.1 (1), pp. 56–71. DOI: 10.1287/ijoc.1110.0493. URL: https://www.iro.umontreal.ca/~lecuyer/myftp/papers/split-static-rel.pdf (visited on 02/11/2019) (cit. on pp. ix, 1, 9, 23, 70).

Bréhier, Charles-Edouard, Ludovic Goudenège, and Loïc Tudela (2016). "Central Limit Theorem for Adaptive Multilevel Splitting Estimators in an Idealized Setting". In: *Monte Carlo and Quasi-Monte Carlo Methods*. Ed. by Ronald Cools and Dirk Nuyens. Vol. 163. Springer Proceedings in Mathematics & Statistics. Cham: Springer International Publishing, pp. 245–260. ISBN: 978-3-319-33507-0. DOI: 10.1007/978-3-319-33507-0_10. URL: http://arxiv.org/abs/1501.01399 (cit. on p. 113).

Bréhier, Charles-Edouard, Tony Lelièvre, and Mathias Rousset (2015). "Analysis of Adaptive Multilevel Splitting Algorithms in an Idealized Case". In: *ESAIM: Probability and Statistics* 19, pp. 361–394. DOI: 10.1051/ps/2014029. arXiv: [objectObject]. URL: http://arxiv.org/abs/1405.1352 (visited on 01/02/2020) (cit. on pp. 23, 35, 40, 113).

Brown, Judith C. (1991). "Calculation of a Constant Q Spectral Transform". In: *The Journal of the Acoustical Society of America* 89.1 (1), pp. 425–434. DOI: 10.1121/1.400476. URL: https://www.ee.columbia.edu/~dpwe/papers/Brown91-cqt.pdf (visited on 10/10/2018) (cit. on p. 132).

Buch, Michael, Elio Quinton, and Bob L Sturm (2017). "NichtnegativeMatrixFaktorisierungnutzendesKlangsynthesenSystem (NiMFKS): Extensions of NMF-Based Concatenative Sound Synthesis". In: *Proceedings of the 20th International Conference on Digital Audio Effects*. 20th International Conference on Digital Audio Effects (DAFx-17). Edinburgh, p. 7 (cit. on pp. 125, 127, 142).

Bucklew, James (2004). *Introduction to Rare Event Simulation*. 2004 edition. New York: Springer. 268 pp. ISBN: 978-0-387-20078-1 (cit. on p. 12).

Caetano, Marcelo and Xavier Rodet (2013). "Musical Instrument Sound Morphing Guided by Perceptually Motivated Features". In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.8 (8), pp. 1666–1675. DOI: 10.1109/TASL.2013.2260154. URL: http://articles.ircam.fr/textes/Caetano13a/index.pdf (cit. on pp. 125, 127, 132).

Cancela, H., G. Rubino, and B. Tuffin (2005). "New Measures of Robustness in Rare Event Simulation". In: *Proceedings of the Winter Simulation Conference, 2005.* Proceedings of the Winter Simulation Conference, 2005. DOI: 10.1109/WSC.2005.1574290 (cit. on p. 15).

Cariani, P. A. and B. Delgutte (1996). "Neural Correlates of the Pitch of Complex Tones. I. Pitch and Pitch Salience." In: *Journal of neurophysiology* 76.3 (3), pp. 1698–1716. DOI: 10.1152/jn.1996.76.3.1698. pmid: 8890286. URL: http://www.brainmusic.org/MBB91%20Webpage/Pitch_II_Cariani.pdf (visited on 10/11/2018) (cit. on p. 133).

Caterini, Anthony L., Arnaud Doucet, and Dino Sejdinovic (2018). "Hamiltonian Variational Auto-Encoder". In: *Advances in Neural Information Processing Systems*. arXiv: 1805.11328. URL: http://arxiv.org/abs/1805.11328 (visited on 10/02/2019) (cit. on p. 113).

Cérou, Frédéric and Arnaud Guyader (2007). "Adaptive Multilevel Splitting for Rare Event Analysis". In: *Stochastic Analysis and Applications* 25.2 (2), pp. 417–443. DOI: 10.1080/07362990601139628. URL: ftp://ftp.idsa.prd.fr/techreports/2005/PI-1747.pdf (visited on 07/09/2019) (cit. on pp. 23, 40, 113).

— (2016). "Fluctuation Analysis of Adaptive Multilevel Splitting". In: *The Annals of Applied Probability* 26.6 (6), pp. 3319–3380. DOI: 10.1214/16-AAP1177. arXiv: 1408.6366 (cit. on pp. 23, 40, 113).

Cérou, Frédéric et al. (2006). "Genetic Genealogical Models in Rare Event Analysis". In: *ALEA, Latin American Journal of Probability and Mathematical Statistics*. URL: https://hal.inria.fr/inria-00071391 (visited on 01/09/2020) (cit. on pp. 23, 24, 34, 38).

Chan, Hock Peng and Tze Leung Lai (2013). "A General Theory of Particle Filters in Hidden Markov Models and Some Applications". In: *The Annals of Statistics* 41.6 (6), pp. 2877–2904. DOI: 10.1214/13-AOS1172. URL: https://projecteuclid.org/euclid.aos/1387313393 (visited on 07/30/2019) (cit. on p. 34).

Charles-Edouard, Bréhier et al. (2015). *Unbiasedness of Some Generalized Adaptive Multilevel Splitting Algorithms*. arXiv: 1505.02674 [math, stat]. URL: http:

//arxiv.org/abs/1505.02674 (visited on 07/09/2019) (cit. on pp. 23, 40, 113).

Chazan, Dan and Ron Hoory (2006). "Feature-Domain Concatenative Speech Synthesis". U.S. pat. 7035791B2. International Business Machines Corp. URL: https://patents.google.com/patent/US7035791B2/en (visited on 02/12/2019) (cit. on p. 126).

Coleman, Graham and Jordi Bonada (2008). "Sound Transformation by Descriptor Using an Analytic Domain". In: *Proceedings of the 11th Int. Conference on Digital Audio Effects (DAFx-08), Espoo, Finland, September 1-4, 2008*. DAFx, p. 7 (cit. on pp. 127, 132).

Coleman, Graham, Esteban Maestre, and Jordi Bonada (2010). "Augmenting Sound Mosaicing with Descriptor-Driven Transformation". In: *Proceedings of DAFx-10*, p. 4 (cit. on pp. 125, 132).

Collins, Nick and Bob L. Sturm (2011). "Sound Cross-Synthesis and Morphing Using Dictionary-Based Methods". In: *International Computer Music Conference*. URL: http://vbn.aau.dk/files/77310007/dbmcrossynth.pdf (cit. on pp. 123, 125).

Davis, Geoffrey M., Stephane G. Mallat, and Zhifeng Zhang (1994). "Adaptive Time-Frequency Decompositions with Matching Pursuit". In: *Wavelet Applications*. Wavelet Applications. Vol. 2242. International Society for Optics and Photonics, pp. 402–414. DOI: 10.1117/12.170041 (cit. on p. 134).

Dean, Thomas and Paul Dupuis (2009). "Splitting for Rare Event Simulation: A Large Deviation Approach to Design and Analysis". In: *Stochastic Processes and their Applications* 119.2 (2), pp. 562–587. DOI: 10.1016/j.spa.2008.02.017. URL: https://www.dam.brown.edu/lcds/publications/documents/Dupuis_SplittingPaper.pdf (visited on 07/08/2020) (cit. on pp. 23, 34).

Del Moral, Pierre (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. 2004 edition. Latheronwheel, Caithness: Springer. 580 pp. ISBN: 978-1-870325-07-3 (cit. on p. 34).

Del Moral, Pierre, Arnaud Doucet, and Ajay Jasra (2006). "Sequential Monte Carlo Samplers". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3 (3), pp. 411–436. DOI: 10.1111/j.1467-9868.2006.00553.x. URL: http://www.stats.ox.ac.uk/~doucet/delmoral_doucet_

jasra_sequentialmontecarlosamplersJRSSB.pdf (visited on 11/24/2014) (cit. on p. 23).

Del Moral, Pierre and Pascal Lezaud (2006). "Branching and Interacting Particle Interpretations of Rare Event Probabilities". In: *Stochastic Hybrid Systems*. Lecture Notes in Control and Information Science, Volume 337. Berlin, Heidelberg: Springer, pp 277–323. ISBN: 978-3-540-33467-5. DOI: 10.1007/11587392_9. URL: https://hal-enac.archives-ouvertes.fr/hal-00968737 (visited on 01/09/2020) (cit. on pp. 23, 24).

Devroye, Luc (1986). *Non-Uniform Random Variate Generation*. New York: Springer. 843 pp. ISBN: 978-0-387-96305-1 (cit. on p. 68).

DiCiccio, Thomas J. and Bradley Efron (1996). "Bootstrap Confidence Intervals". In: *Statistical Science* 11.3 (3), pp. 189–212. DOI: 10.1214/ss/1032280214. URL: https://statistics.stanford.edu/sites/default/files/BIO%20175.pdf (visited on 02/05/2015) (cit. on p. 19).

Domke, Justin (2020). *Moment-Matching Conditions for Exponential Families with Conditioning or Hidden Data*. arXiv: 2001.09771 [cs, stat]. URL: http://arxiv.org/abs/2001.09771 (visited on 06/29/2020) (cit. on p. 4).

Doucet, Arnaud, Nando Freitas, and Neil Gordon (2001). *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer New York. ISBN: 978-1-4757-3437-9. URL: http://public.eblib.com/choice/publicfullrecord.aspx?p=3087052 (visited on 09/01/2017) (cit. on p. 23).

Driedger, Jonathan and Meinard Müller (2016). "A Review of Time-Scale Modification of Music Signals". In: *Applied Sciences* 6.2 (2), p. 57. DOI: 10.3390/app6020057. URL: https://www.mdpi.com/2076-3417/6/2/57 (visited on 02/26/2019) (cit. on pp. 125, 126, 134).

Driedger, Jonathan and Thomas Pratzlich (2015). "Let It Bee – Towards NMF-Inspired Audio Mosaicing". In: *Proceedings of ISMIR*. International Society for Music Information Retrieval. Malaga, p. 7. URL: http://ismir2015.uma.es/articles/13_Paper.pdf (cit. on pp. 125, 127).

Dudley, Homer (1964). "Thirty Years of Vocoder Research". In: *The Journal of the Acoustical Society of America* 36.5 (5), pp. 1021–1021. DOI: 10.1121/1.2143221 (cit. on p. 127).

Efron, B (1979). "Bootstrap Methods: Another Look at the Jackknife". In: *The Annals of Statistics* 7.1 (1), pp. 1–26. DOI: 10.1214/aos/1176344552 (cit. on pp. 19, 26).

Elowsson, Anders and Anders Friberg (2017). "Long-Term Average Spectrum in Popular Music and Its Relation to the Level of the Percussion". In: *Audio Engineering Society Convention 142*. Audio Engineering Society, p. 13 (cit. on p. 137).

Embrechts, Paul, S Kluppelberg, and Thomas Mikosch (1997). *Extremal Events in Finance and Insurance*. Springer Berlin Heidelberg (cit. on pp. 83, 86).

Embrechts, Paul, Filip Lindskog, and Alexander J McNeil (2003). "Modelling Dependence with Copulas and Applications to Risk Management". In: *Handbook of heavy tailed distributions in finance* 8.329-384 (329-384), p. 1. URL: https://people.math.ethz.ch/~embrecht/ftp/copchapter.pdf (cit. on p. 70).

Engel, Jesse et al. (2017). "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders". In: *PMLR*. International Conference on Machine Learning. arXiv: 1704.01279. URL: http://arxiv.org/abs/1704.01279 (cit. on p. 127).

Ferguson, Thomas S. (1974). "Prior Distributions on Spaces of Probability Measures". In: *The Annals of Statistics* 2.4 (4), pp. 615–629. DOI: 10.1214/aos/1176342752. URL: https://projecteuclid.org/euclid.aos/1176342752 (visited on 02/12/2020) (cit. on p. 118).

Ferguson, Thomas S. and Michael J. Klass (1972). "A Representation of Independent Increment Processes without Gaussian Components". In: *The Annals of Mathematical Statistics* 43.5 (5), pp. 1634–1643. DOI: 10.1214/aoms/1177692395. URL: https://projecteuclid.org/euclid.aoms/1177692395 (visited on 02/12/2020) (cit. on p. 118).

Fisher, R. A. and L. H. C. Tippett (1928). "Limiting Forms of the Frequency Distribution of the Largest or Smallest Member of a Sample". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24.2 (2), pp. 180–190. DOI: 10.1017/S0305004100015681 (cit. on p. 83).

Garvels, M. J. J. (2000). "The splitting method in rare event simulation". URL: https://research.utwente.nl/en/publications/the-splitting-

method-in-rare-event-simulation (visited on 09/04/2019) (cit. on pp. 23, 35, 37, 39).

Garvels, M. J. J. and D. P. Kroese (1998). "A Comparison of RESTART Implementations". In: *Proceedings of the 1998 Winter Simulation Conference*. 1998 Winter Simulation Conference. Vol. 1. Washington, DC, USA: IEEE, 601–608 vol.1. ISBN: 978-0-7803-5133-2. DOI: 10.1109/WSC.1998.745040 (cit. on pp. 34, 40).

Garvels, Marnix J. J., Jan-Kees C. W. Van Ommeren, and Dirk P. Kroese (2002). "On the Importance Function in Splitting Simulation". In: *European Transactions on Telecommunications* 13.4 (4), pp. 363–371. DOI: 10.1002/ett.4460130408 (cit. on pp. 34, 40).

Geman, Stuart and Donald Geman (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (6), pp. 721–741. DOI: 10.1109/TPAMI.1984.4767596. pmid: 22499653. URL: http://www.dam.brown.edu/people/documents/stochasticrelaxation.pdf (cit. on p. 20).

Gertsbakh, Ilya B. and Yoseph Shpungin (2016). *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. 1st. USA: CRC Press. 221 pp. ISBN: 978-1-4398-1742-1. Google Books: 3tF8216VlG8C (cit. on p. 70).

Glasserman, P. et al. (1998a). "A Large Deviations Perspective on the Efficiency of Multilevel Splitting". In: *IEEE Transactions on Automatic Control* 43.12 (12), pp. 1666–1679. DOI: 10.1109/9.736061. URL: https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/4274/deviations_perspective.pdf (visited on 12/31/2019) (cit. on pp. 23, 34).

Glasserman, Paul et al. (1998b). "A Look At Multilevel Splitting". In: *Monte Carlo and Quasi-Monte Carlo Methods 1996*. Ed. by Harald Niederreiter et al. Lecture Notes in Statistics. New York, NY: Springer, pp. 98–108. ISBN: 978-1-4612-1690-2. DOI: 10.1007/978-1-4612-1690-2_5 (cit. on p. 23).

— (1999). "Multilevel Splitting for Estimating Rare Event Probabilities". In: *Operations Research* 47.4 (4), pp. 585–600. DOI: 10.1287/opre.47.4.585. URL: https://www0.gsb.columbia.edu/mygsb/faculty/research/pubfiles/4273/multilevel_splitting.pdf (visited on 04/06/2020) (cit. on pp. 34, 35, 40).

Glynn, Peter W. and Ward Whitt (1992). "The Asymptotic Efficiency of Simulation Estimators". In: *Operations Research* 40.3 (3), pp. 505–520. DOI: 10.1287/opre.40.3.505. URL: http://www.columbia.edu/~ww2040/efficiency.pdf (visited on 03/23/2020) (cit. on p. 17).

Goodwin, Michael (1997). "Matching Pursuit with Damped Sinusoids". In: *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 3. Munich, Germany: IEEE, pp. 2037–2040. ISBN: 978-0-8186-7919-3. DOI: 10.1109/ICASSP.1997.599345. URL: https://www2.spsc.tugraz.at/people/franklyn/ICASSP97/pdf/author/ic972037.pdf (visited on 04/01/2016) (cit. on pp. 136, 138).

Green, P. J. (1984). "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and Some Robust and Resistant Alternatives". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 46.2, pp. 149–192. JSTOR: 2345503 (cit. on p. 86).

Griewank, Andreas and Andrea Walther (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics. 438 pp. ISBN: 978-0-89871-659-7 (cit. on p. 85).

Griffin, D. and Jae Lim (1984). "Signal Estimation from Modified Short-Time Fourier Transform". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32.2 (2), pp. 236–243. DOI: 10.1109/TASSP.1984.1164317. URL: http://musicweb.ucsd.edu/~sdubnov/CATbox/Reader/GriffinLimMSTFT.pdf (cit. on pp. 126, 132).

Grimshaw, Scott D. (1993). "Computing Maximum Likelihood Estimates for the Generalized Pareto Distribution". In: *Technometrics* 35.2 (2), pp. 185–191. DOI: 10.1080/00401706.1993.10485040 (cit. on p. 86).

Grinstein, Eric et al. (2017). "Audio Style Transfer". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 586–590. DOI: 10.1109/ICASSP.2018.8461711. arXiv: 1710.11385. URL: http://arxiv.org/abs/1710.11385 (visited on 01/14/2018) (cit. on p. 127).

Gudmundsson, Thorbjörn and Henrik Hult (2014). "Markov Chain Monte Carlo for Computing Rare-Event Probabilities for a Heavy-Tailed Random Walk". In: *Journal of Applied Probability* 51.2 (2), pp. 359–376. DOI: 10.1239/jap/1402578630 (cit. on pp. 20, 75).

Guyader, Arnaud, Nicolas Hengartner, and Eric Matzner-Løber (2011). "Simulation and Estimation of Extreme Quantiles and Extreme Probabilities". In: *Applied Mathematics & Optimization* 64.2 (2), pp. 171–196. DOI: 10.1007/s00245-011-9135-z. URL: http://www.lpsm.paris/pageperso/guyader/files/papers/ghm.pdf (visited on 07/10/2020) (cit. on pp. 35, 112).

Hagstrom, Jane N. (1990). "Computing the Probability Distribution of Project Duration in a PERT Network". In: *Networks* 20.2 (2), pp. 231–244. DOI: 10.1002/net.3230200208 (cit. on p. 70).

Hoffman, Matt and Perry R Cook (2006). "Feature-Based Synthesis: A Tool for Evaluating, Designing, and Interacting with Music IR Systems". In: *Proceedings of ISMIR*, p. 2. URL: http://soundlab.cs.princeton.edu/publications/2006_ismir_fbs.pdf (cit. on p. 125).

Hoffman, Matthew D., Perry R. Cook, and David M. Blei (2009). "Bayesian Spectral Matching: Turning Young MC into MC Hammer via MCMC Sampling." In: *ICMC* (cit. on pp. 125, 132, 147).

Holland, Paul W. and Roy E. Welsch (1977). "Robust Regression Using Iteratively Reweighted Least-Squares". In: *Communications in Statistics - Theory and Methods* 6.9, pp. 813–827. DOI: 10.1080/03610927708827533 (cit. on p. 86).

Hosking, J. R. M. and J. R. Wallis (1987). "Parameter and Quantile Estimation for the Generalized Pareto Distribution". In: *Technometrics* 29.3 (3), pp. 339–349. DOI: 10.1080/00401706.1987.10488243 (cit. on pp. 86, 112).

Hu, Feifang and James V. Zidek (2002). "The Weighted Likelihood". In: *The Canadian Journal of Statistics / La Revue Canadienne de Statistique* 30.3 (3), pp. 347–371. DOI: 10.2307/3316141 (cit. on p. 86).

Hüsler, Jürg, Deyuan Li, and Mathias Raschke (2011). "Estimation for the Generalized Pareto Distribution Using Maximum Likelihood and Goodness of Fit". In: *Communications in Statistics - Theory and Methods* 40.14 (14), pp. 2500–2510. DOI: 10.1080/03610920903324874. URL: https://doi.org/10.1080/03610920903324874 (visited on 05/31/2020) (cit. on p. 86).

Jaganathan, Kishore, Yonina C. Eldar, and Babak Hassibi (2015). *Phase Retrieval: An Overview of Recent Developments*. arXiv: 1510.07713 [cs, math]. URL: http://arxiv.org/abs/1510.07713 (visited on 03/06/2019) (cit. on p. 134).

Johansen, Adam M., Pierre Del Moral, and Arnaud Doucet (2006). "Sequential Monte Carlo Samplers for Rare Events". In: *Proceedings of the 6th International Workshop on Rare Event Simulation*, pp. 256–267. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.61.2888 (visited on 02/11/2015) (cit. on pp. 23, 24).

Kahn, Herman and Theodore E. Harris (1951). "Estimation of Particle Transmission by Random Sampling". In: *National Bureau of Standards applied mathematics series* 12, pp. 27–30 (cit. on pp. 23, 28).

Kemna, A. G. Z. and A. C. F. Vorst (1990). "A Pricing Method for Options Based on Average Asset Values". In: *Journal of Banking & Finance* 14.1 (1), pp. 113–129. URL: https://ideas.repec.org/a/eee/jbfina/v14y1990i1p113-129.html (visited on 07/31/2020) (cit. on p. 74).

Kong, A. (2014). "Importance Sampling". In: *Wiley StatsRef: Statistics Reference Online*. American Cancer Society. ISBN: 978-1-118-44511-2. DOI: 10.1002/9781118445112.stat05402. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118445112.stat05402 (visited on 06/08/2020) (cit. on p. 21).

Kriman, V. and R. Y. Rubinstein (1995). "Polynomial Time Algorithms for Estimation of Rare Events in Queueing Models". In: *Frontiers in Queueing: Models and Applications in Science and Engineering*. Ed. by Jewgeni H. Dshalalow. CRC Press. ISBN: 978-0-8493-8076-1. Google Books: t4S4i8HkJXkC. URL: https://core.ac.uk/download/pdf/6377029.pdf (visited on 07/28/2020) (cit. on p. 15).

Kroese, Dirk P., Thomas Taimre, and Zdravko I. Botev (2011). *Handbook of Monte Carlo Methods*. Wiley Series in Probability and Statistics 706. Hoboken, N.J: Wiley. 743 pp. ISBN: 978-0-470-17793-8 (cit. on pp. 12, 20, 22, 24, 28).

Kyprianou, Andreas E. (2014). *Fluctuations of Lévy Processes with Applications: Introductory Lectures*. Second edition. Universitext. Heidelberg: Springer. 455 pp. ISBN: 978-3-642-37631-3 (cit. on pp. 50, 117).

L'Ecuyer, Pierre, Zdravko I. Botev, and Dirk P. Kroese (2018). "On a Generalized Splitting Method for Sampling from a Conditional Distribution". In: *2018 Winter Simulation Conference (WSC)*. 2018 Winter Simulation Conference (WSC). IEEE Press. Gothenburg, Sweden: IEEE, pp. 1694–1705. ISBN: 978-1-5386-6572-5. DOI: 10.1109/WSC.2018.8632422. URL: http://www.iro.umontreal.ca/~lecuyer/myftp/papers/wsc18splitcond.pdf (visited on 09/08/2020) (cit. on pp. 23, 38).

L'Ecuyer, Pierre, Valérie Demers, and Bruno Tuffin (2006). "Splitting for Rare-Event Simulation". In: *38th Conference on Winter Simulation*. Proceedings of the 2006 Winter Simulation Conference. WSC '06. Monterey, California: Winter Simulation Conference, pp. 137–148. ISBN: 978-1-4244-0501-5. DOI: 10.1109/WSC.2006.323046 (cit. on pp. 34, 35).

L'Ecuyer, Pierre et al. (2009). "Splitting Techniques". In: *Rare Event Simulation Using Monte Carlo Methods*. John Wiley & Sons, Ltd, Chapter 3. ISBN: 978-0-470-74540-3. DOI: 10.1002/9780470745403.ch3 (cit. on pp. 23, 34, 37).

L'Ecuyer, Pierre et al. (2010). "Asymptotic Robustness of Estimators in Rare-Event Simulation". In: *ACM Transactions on Modeling and Computer Simulation* 20.1 (1), 6:1–6:41. DOI: 10.1145/1667072.1667078. URL: https://hal.archives-ouvertes.fr/inria-00170077/ (visited on 03/20/2020) (cit. on p. 15).

Lagnoux, Agnes (2006). "Rare Event Simulation". In: *Probability in the Engineering and Informational Sciences* 20.1 (1), pp. 45–66. DOI: 10.1017/S0269964806060025. URL: http://www.math.univ-toulouse.fr/~lagnoux/articlePEIS.pdf (visited on 07/13/2020) (cit. on pp. 35, 37).

Langner, Gerald (1992). "Periodicity Coding in the Auditory System". In: *Hearing Research* 60.2 (2), pp. 115–142. DOI: 10.1016/0378-5955(92)90015-F (cit. on p. 133).

Lattner, Stefan, Monika Dorfler, and Andreas Arzt (2019). "Learning Complex Basis Functions for Invariant Representations of Audio". In: *Proceedings of the 20th Conference of the International Society for Music Information Retrieval*. ISMIR 2019, p. 8. arXiv: 1907.05982. URL: http://archives.ismir.net/ismir2019/paper/000085.pdf (cit. on p. 133).

Le Gland, François and Nadia Oudjane (2006). "A Sequential Particle Algorithm That Keeps the Particle System Alive". In: *Stochastic Hybrid Systems*. Ed. by Henk A. P. Blom and John Lygeros. Vol. 337. Lecture Notes in Control and Information Science. Berlin/Heidelberg: Springer-Verlag, pp. 351–389. ISBN: 978-3-540-33466-8. DOI: 10.1007/11587392_11. URL: http://link.springer.com/10.1007/11587392_11 (visited on 07/08/2020) (cit. on p. 24).

Licklider, J. C. R. (1951). "A Duplex Theory of Pitch Perception". In: *Experientia* 7.4 (4), pp. 128–134. DOI: 10.1007/BF02156143. URL: http://web.mit.edu/HST.723/www/ThemePapers/Pitch/Licklider1951.pdf (visited on 01/17/2019) (cit. on p. 133).

Limpert, Eckhard, Werner A. Stahel, and Markus Abbt (2001). "Log-Normal Distributions across the Sciences: Keys and Clues". In: *BioScience* 51.5 (5), pp. 341–352. DOI: 10.1641/0006-3568(2001)051[0341:LNDATS]2.0.CO;2. URL: https://stat.ethz.ch/~stahel/lognormal/bioscience2001.pdf (visited on 06/02/2020) (cit. on p. 62).

Liutkus, Antoine et al. (2014). "Kernel Spectrogram Models for Source Separation". In: IEEE, pp. 6–10. ISBN: 978-1-4799-3109-5. DOI: 10.1109/HSCMA.2014.6843240 (cit. on p. 144).

Luo, Yin-Jyun, Kat Agres, and Dorien Herremans (2019). "Learning Disentangled Representations of Timbre and Pitch for Musical Instrument Sounds Using Gaussian Mixture Variational Autoencoders". In: *Proceedings of the 20th Conference of the International Society for Music Information Retrieval*. ISMIR 2019. arXiv: 1906.08152. URL: http://arxiv.org/abs/1906.08152 (visited on 10/24/2019) (cit. on p. 133).

MacKinlay, Daniel and Zdravko I Botev (2019). "Mosaic Style Transfer Using Sparse Autocorrelograms". In: *Proceedings of the 20th Conference of the International Society for Music Information Retrieval*. ISMIR 2019. Delft, p. 5. URL: http://archives.ismir.net/ismir2019/paper/000109.pdf (cit. on pp. xi, 123, 131).

Makarov, Mikhail (2006). "Extreme Value Theory and High Quantile Convergence". In: *The Journal of Operational Risk* 1.2 (2), pp. 51–57. DOI: 10.21314/JOP.2006.009. URL: http://www.risk.net/journal-of-operational-

risk/technical-paper/2160852/extreme-value-theory-quantile-convergence (visited on 12/20/2019) (cit. on pp. 91, 112).

Markovitch, Natalia M and Udo R Krieger (2002). "The Estimation of Heavy-Tailed Probability Density Functions, Their Mixtures and Quantiles". In: *Computer Networks* 40.3 (3), pp. 459–474. DOI: 10.1016/S1389-1286(02)00306-7 (cit. on pp. 86, 112).

McNeil, Alexander J, Rüdiger Frey, and Paul Embrechts (2005). *Quantitative Risk Management : Concepts, Techniques and Tools*. Princeton: Princeton Univ. Press. ISBN: 0-691-12255-5 (cit. on pp. 9, 11, 82).

McNeil, Alexander J. (1997). "Estimating the Tails of Loss Severity Distributions Using Extreme Value Theory". In: *ASTIN Bulletin: The Journal of the IAA* 27.1 (1), pp. 117–137. DOI: 10.2143/AST.27.1.563210. URL: http://www.financerisks.com/filedati/WP/EVT/astin.pdf (visited on 12/20/2019) (cit. on pp. 82, 86, 112).

Mermelstein, Paul and CH Chen (1976). "Distance Measures for Speech Recognition: Psychological and Instrumental". In: *Pattern Recognition and Artificial Intelligence,* vol. 101. Academic Press, pp. 374–388. URL: http://web.haskins.yale.edu/sr/SR047/SR047_07.pdf (cit. on p. 132).

Mogensen, Patrick K. and Asbjørn N. Riseth (2018). "Optim: A Mathematical Optimization Package for Julia". In: *Journal of Open Source Software* 3.24 (24), p. 615. DOI: 10.21105/joss.00615 (cit. on p. 85).

Mogensen, Patrick Kofod et al. (2020). *JuliaNLSolvers/Optim.Jl: V0.22.0*. Version v0.22.0. Zenodo. DOI: 10.5281/ZENODO.3909570. URL: https://zenodo.org/record/3909570 (visited on 07/06/2020) (cit. on p. 85).

Mohamed, Shakir et al. (2020). "Monte Carlo Gradient Estimation in Machine Learning". In: *Journal of Machine Learning Research* 21.132, pp. 1–62. arXiv: 1906.10652. URL: http://jmlr.org/papers/v21/19-346.html (visited on 09/17/2020) (cit. on p. 113).

Nam, S. S., M.-S. Alouini, and H. Yang (2010). "An MGF-Based Unified Framework to Determine the Joint Statistics of Partial Sums of Ordered Random Variables". In: *IEEE Transactions on Information Theory* 56.11 (11), pp. 5655–5672. DOI: 10.1109/TIT.2010.2070271 (cit. on p. 64).

Nam, S. S., Y. Ko, and M.-S. Alouini (2017). "New Closed-Form Results on Ordered Statistics of Partial Sums of Gamma Random Variables and Its Application to Performance Evaluation in the Presence of Nakagami Fading". In: *IEEE Access* 5, pp. 12820–12832. DOI: 10.1109/ACCESS.2017.2717048 (cit. on p. 64).

Pati, Y. C., R. Rezaiifar, and P. S. Krishnaprasad (1993). "Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition". In: *Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*. The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers, 40–44 vol.1. DOI: 10.1109/ACSSC.1993.342465 (cit. on p. 134).

Perraudin, Nathanael, Peter Balazs, and Peter L. Sondergaard (2013). "A Fast Griffin-Lim Algorithm". In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. 2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2013). New Paltz, NY: IEEE, pp. 1–4. ISBN: 978-1-4799-0972-8. DOI: 10.1109/WASPAA.2013.6701851 (cit. on p. 126).

Pickands III, James (1975). "Statistical Inference Using Extreme Order Statistics". In: *The Annals of Statistics* 3.1 (1), pp. 119–131. DOI: 10.1214/aos/1176343003. URL: https://projecteuclid.org/euclid.aos/1176343003 (visited on 12/24/2019) (cit. on p. 83).

Prony, R. (1795). "Essai Éxperimental et Analytique: Sur Les Lois de La Dilatabilité de Fluides Élastique et Sur Celles de La Force Expansive de La Vapeur de l'alkool, à Différentes Températures". In: *Journal de l'École Polytechnique Floréal et Plairial* 2. URL: http://users.polytech.unice.fr/~leroux/PRONY.pdf (cit. on p. 138).

Rabiner, L. (1977). "On the Use of Autocorrelation Analysis for Pitch Detection". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.1 (1), pp. 24–33. DOI: 10.1109/TASSP.1977.1162905 (cit. on p. 133).

Rall, Louis B. (1981). *Automatic Differentiation: Techniques and Applications*. Lecture Notes in Computer Science 120. Berlin ; New York: Springer-Verlag. 165 pp. ISBN: 978-3-540-10861-0 (cit. on p. 85).

Rasmussen, Carl Edward and Christopher K. I. Williams (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, p. 248. 248 pp. ISBN: 978-0-262-18253-9. URL: http://www.gaussianprocess.org/gpml/ (cit. on p. 143).

Revels, Jarrett, Miles Lubin, and Theodore Papamarkou (2016). *Forward-Mode Automatic Differentiation in Julia*. arXiv: 1607.07892 [cs]. URL: http://arxiv.org/abs/1607.07892 (visited on 02/05/2019) (cit. on p. 85).

Roads, Curtis (2004). *Microsound*. Cambridge, Mass.: The MIT Press. 424 pp. ISBN: 978-0-262-68154-4 (cit. on p. 126).

Robert, Christian P. (1995). "Simulation of Truncated Normal Variables". In: *Statistics and Computing* 5.2 (2), pp. 121–125. DOI: 10.1007/BF00143942. arXiv: 0907.4010. URL: http://arxiv.org/abs/0907.4010 (visited on 07/31/2020) (cit. on p. 74).

Robert, Christian P. et al. (2018). "Accelerating MCMC Algorithms". In: *WIREs Computational Statistics* 10.5 (5), e1435. DOI: 10.1002/wics.1435. arXiv: 1804.02719. URL: http://arxiv.org/abs/1804.02719 (visited on 04/28/2020) (cit. on p. 113).

Roberts, G.O. and A.F.M. Smith (1994). "Simple Conditions for the Convergence of the Gibbs Sampler and Metropolis-Hastings Algorithms". In: *Stochastic Processes and their Applications* 49.2 (2), pp. 207–216. DOI: 10.1016/0304-4149(94)90134-1 (cit. on p. 21).

Rubin, Donald B. (1987). "The Calculation of Posterior Distributions by Data Augmentation: Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputations When Fractions of Missing Information Are Modest: The SIR Algorithm". In: *Journal of the American Statistical Association* 82.398, pp. 543–546. DOI: 10.2307/2289460 (cit. on p. 21).

Rubino, Gerardo and Bruno Tuffin, eds. (2009). *Rare Event Simulation Using Monte Carlo Methods*. 1st ed. Chichester, U.K: John Wiley & Sons, Ltd. 268 pp. ISBN: 978-0-470-77269-0. DOI: 10.1002/9780470745403 (cit. on pp. 12, 21, 22, 24, 28).

Rubinstein, Reuven Y. and Dirk P. Kroese (2016). *Simulation and the Monte Carlo Method*. 3 edition. Wiley Series in Probability and Statistics. Hoboken, New

Jersey: Wiley. 432 pp. ISBN: 978-1-118-63216-1 (cit. on pp. 12, 15, 20, 24, 28, 116, 118).

Salimans, Tim, Diederik Kingma, and Max Welling (2015). "Markov Chain Monte Carlo and Variational Inference: Bridging the Gap". In: *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. International Conference on Machine Learning. ICML'15. Lille, France: JMLR.org, pp. 1218–1226. URL: http://proceedings.mlr.press/v37/salimans15.html (visited on 08/30/2017) (cit. on p. 113).

Sato, Ken-iti (1999). *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press. 502 pp. ISBN: 978-0-521-55302-5 (cit. on pp. 50, 117).

Schwarz, Diemo (2011). "State of the Art in Sound Texture Synthesis". In: *Proceedings of DAFx-11*, pp. 221–231. URL: http://recherche.ircam.fr/pub/dafx11/Papers/30_e.pdf (visited on 12/08/2015) (cit. on p. 125).

Serra, Xavier and Julius Smith (1990). "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition". In: *Computer Music Journal* 14.4 (4), pp. 12–24. DOI: 10.2307/3680788. URL: http://www.mtg.upf.edu/node/251 (cit. on p. 138).

Shahabuddin, Perwez (1994). "Importance Sampling for the Simulation of Highly Reliable Markovian Systems". In: *Management Science* 40.3 (3), pp. 333–352. DOI: 10.1287/mnsc.40.3.333 (cit. on p. 15).

Shechtman, Y. et al. (2015). "Phase Retrieval with Application to Optical Imaging: A Contemporary Overview". In: *IEEE Signal Processing Magazine* 32.3 (3), pp. 87–109. DOI: 10.1109/MSP.2014.2352673. URL: http://arxiv.org/abs/1402.7350 (cit. on p. 134).

Simon, Ian et al. (2005). "Audio Analogies: Creating New Music from an Existing Performance by Concatenative Synthesis". In: *Proceedings of the 2005 International Computer Music Conference*, pp. 65–72. URL: http://research.microsoft.com/en-us/um/redmond/groups/cue/compmusic/audioanalogies_icmc2005.pdf (visited on 01/16/2017) (cit. on p. 125).

Simon, Marvin Kenneth and Mohamed-Slim Alouini (2005). *Digital Communication over Fading Channels*. 2nd ed. Wiley Series in Telecommunications and Signal Processing. Hoboken, N.J: Wiley-Interscience. 900 pp. ISBN: 978-0-471-64953-3 (cit. on pp. 10, 62).

Slaney, M., M. Covell, and B. Lassiter (1996). "Automatic Audio Morphing". In: *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. On Conference Proceedings., 1996 IEEE International Conference - Volume 02*. 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. Vol. 2. ICASSP '96. Washington, DC, USA: IEEE Computer Society, pp. 1001–1004. ISBN: 978-0-7803-3192-1. DOI: 10.1109/ICASSP.1996.543292 (cit. on p. 127).

Slaney, M. and R. F. Lyon (1990). "A Perceptual Pitch Detector". In: *Proceedings of ICASSP*. International Conference on Acoustics, Speech, and Signal Processing, 357–360 vol.1. DOI: 10.1109/ICASSP.1990.115684. URL: http://www.dicklyon.com/tech/Hearing/PerceptualPitch-SlaneyLyon.pdf (cit. on p. 133).

Slaney, M., D. Naar, and R.E. Lyon (1994). "Auditory Model Inversion for Sound Separation". In: *Proceedings of ICASSP '94*. ICASSP '94. IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. ii. Adelaide, SA, Australia: IEEE, pp. II/77–II/80. ISBN: 978-0-7803-1775-8. DOI: 10.1109/ICASSP.1994.389714 (cit. on p. 128).

Smith, Julius O. (2018). *Digital Audio Resampling Home Page*. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University. URL: https://ccrma.stanford.edu/~jos/resample/ (visited on 10/28/2018) (cit. on p. 139).

Smith, Richard L. (1985). "Maximum Likelihood Estimation in a Class of Nonregular Cases". In: *Biometrika* 72.1 (1), pp. 67–90. DOI: 10.2307/2336336. URL: https://academic.oup.com/biomet/article/72/1/67/242523 (visited on 05/31/2020) (cit. on p. 87).

Sondhi, M. (1968). "New Methods of Pitch Extraction". In: *IEEE Transactions on Audio and Electroacoustics* 16.2 (2), pp. 262–266. DOI: 10.1109/TAU.1968.1161986 (cit. on p. 133).

Street, James O., Raymond J. Carroll, and David Ruppert (1988). "A Note on Computing Robust Regression Estimates via Iteratively Reweighted Least Squares". In: *The American Statistician* 42.2, pp. 152–154. DOI: 10.1080/00031305.1988.10475548 (cit. on p. 86).

Sturm, Bob L. (2004). "MATConcat: An Application for Exploring Concatenative Sound Synthesis Using MATLAB." In: *ICMC* (cit. on pp. 123, 125).

Sturm, Bob L. et al. (2009). "Analysis, Visualization, and Transformation of Audio Signals Using Dictionary-Based Methods". In: *Journal of New Music Research* 38.4 (4), pp. 325–341. DOI: 10.1080/09298210903171178 (cit. on p. 125).

Tan, Yue, Yingdong Lu, and Cathy Xia (2018). *Relative Error of Scaled Poisson Approximation via Stein's Method*. arXiv: 1810.04300 [math]. URL: http://arxiv.org/abs/1810.04300 (visited on 05/06/2019) (cit. on p. 63).

Thickstun, John et al. (2017). "MIREX 2017: Frequency Domain Convolutions for Multiple F0 Estimation". In: p. 1 (cit. on p. 133).

Thickstun, John et al. (2018). "Invariances and Data Augmentation for Supervised Music Transcription". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2241–2245. DOI: 10.1109/ICASSP.2018.8461686. arXiv: 1711.04845 (cit. on p. 133).

Vehtari, Aki et al. (2019). *Pareto Smoothed Importance Sampling*. arXiv: 1507.02646 [stat]. URL: http://arxiv.org/abs/1507.02646 (visited on 08/19/2019) (cit. on p. 22).

Verhelst, Werner and Marc Roelands (1993). "An Overlap-Add Technique Based on Waveform Similarity (WSOLA) for High Quality Time-Scale Modification of Speech". In: *Proceedings of ICASSP*. 1993 IEEE International Conference on Acoustics, Speech, and Signal Processing (Minneapolis, Minnesota, USA). ICASSP'93. Washington, DC, USA: IEEE Computer Society, pp. 554–557. ISBN: 978-0-7803-0946-3. DOI: 10.1109/ICASSP.1993.319366. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.202.5460&rep=rep1&type=pdf (visited on 03/04/2019) (cit. on pp. 125, 134).

Verma, Prateek and Julius O. Smith (2018). "Neural Style Transfer for Audio Spectograms". In: *31st Conference on Neural Information Processing Systems (NIPS 2017)*. arXiv: 1801.01589. URL: http://arxiv.org/abs/1801.01589 (visited on 10/06/2018) (cit. on p. 127).

Villén-Altamirano, M. et al. (1994). "Enhancement of the Accelerated Simulation Method RESTART by Considering Multiple Thresholds". In: *Teletraffic Science and Engineering*. Ed. by JACQUES Labetoulle and JAMES W. Roberts.

Vol. 1. The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks. Elsevier, pp. 797–810. DOI: 10.1016/B978-0-444-82031-0.50084-6. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780444820310500846 (visited on 07/11/2019) (cit. on p. 40).

Villén-Altamirano, Manuel and José Villén-Altamirano (1991). "RESTART: A Method for Accelerating Rare Event Simulations". In: *Queueing, performance and Control in ATM*, pp. 71–76. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.452.8093&rep=rep1&type=pdf (cit. on pp. 23, 28).

— (1994). "RESTART: A Straightforward Method for Fast Simulation of Rare Events". In: *Proceedings of the 26th Conference on Winter Simulation*. Proceedings of Winter Simulation Conference. WSC '94. Orlando, Florida, USA: Society for Computer Simulation International, pp. 282–289. ISBN: 978-0-7803-2109-0. DOI: 10.1109/WSC.1994.717150 (cit. on p. 23).

Wang, Steven Xiaogang (2001). "Maximum Weighted Likelihood Estimation". DOI: 10.14288/1.0090880. URL: https://open.library.ubc.ca/collections/ubctheses/831/items/1.0090880 (cit. on p. 86).

Wiener, Norbert (1930). "Generalized Harmonic Analysis". In: *Acta Mathematica* 55, pp. 117–258. DOI: 10.1007/BF02546511. URL: https://projecteuclid.org/euclid.acta/1485887877 (visited on 02/04/2019) (cit. on p. 126).

Wilkinson, William J. et al. (2019). "Unifying Probabilistic Models for Time-Frequency Analysis". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3352–3356. DOI: 10.1109/ICASSP.2019.8682306. URL: http://www.eecs.qmul.ac.uk/~josh/documents/2019/08682306.pdf (cit. on p. 144).

Yaglom, A. M. (1987). *Correlation Theory of Stationary and Related Random Functions. Volume II: Supplementary Notes and References*. Springer Series in Statistics. New York, NY: Springer Science & Business Media. 267 pp. ISBN: 978-1-4612-4628-2 (cit. on p. 143).

Zils, A and F Pachet (2001). "Musical Mosaicing". In: *Proceedings of DAFx-01*. Vol. 2. Limerick, Ireland, p. 135. URL: http://csl.sony.fr/downloads/papers/2001/zils-dafx2001.pdf (cit. on p. 125).