Computability, inference and modeling in probabilistic programming

by

Daniel M. Roy

S.B., Massachusetts Institute of Technology (2004)
M.Eng., Massachusetts Institute of Technology (2006)

Submitted to the Department of
Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2011

Author .....
Department of
Electrical Engineering and Computer Science
April 1, 2011

Certified by .....
Leslie P. Kaelbling
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Computability, inference and modeling in probabilistic programming

by

Daniel M. Roy

Submitted to the Department of
Electrical Engineering and Computer Science
on April 1, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract. We investigate the class of computable probability distributions and explore the fundamental limitations of using this class to describe and compute conditional distributions. In addition to proving the existence of noncomputable conditional distributions, and thus ruling out the possibility of generic probabilistic inference algorithms (even inefficient ones), we highlight some positive results showing that posterior inference is possible in the presence of additional structure like exchangeability and noise, both of which are common in Bayesian hierarchical modeling.

This theoretical work bears on the development of *probabilistic programming languages* (which enable the specification of complex probabilistic models) and their implementations (which can be used to perform Bayesian reasoning). The probabilistic programming approach is particularly well suited for defining infinite-dimensional, recursively-defined stochastic processes of the sort used in nonparametric Bayesian statistics. We present a new construction of the Mondrian process as a partition-valued Markov process in continuous time, which can be viewed as placing a distribution on an infinite kd-tree data structure.

Thesis Supervisor: Leslie P. Kaelbling
Title: Professor of Computer Science and Engineering

Front matter

Acknowledgments

I have enjoyed my time at MIT immensely and will always regard it as a great privilege to have studied and worked here. I would like to thank everyone who helped to give me this opportunity, as well as those whose presence made it so enjoyable. Below, I highlight a few people who have played especially important roles.

I would like to start by thanking my advisor, Leslie Kaelbling, who has been an incredible mentor and unwavering advocate. Leslie encouraged me to pursue my own interests and guided me with her insightful questions and suggestions during our conversations, which always led me to a deeper understanding of my work. My research—as well as my ability to describe it to others—has improved greatly as a result of her perceptive feedback.

I also owe an enormous debt of gratitude to Josh Tenenbaum for his mentorship and support over the course of my graduate career. I have benefitted tremendously from the unparalleled collaborative environment that he has fostered within his lab. Much of the work presented in this dissertation can be traced back to my collaborations with Josh and his students.

I would also like to thank Yee Whye Teh and Scott Aaronson for serving on my doctoral committee along with Leslie and Josh. Yee Whye Teh has been a long-time collaborator: our discovery of the Mondrian process in 2007 was an exciting time and I am very much looking forward to future collaborations. Likewise, I am thankful to Scott for his enthusiasm and for inspiring me to be less apologetic about my theoretical interests.

I must thank Martin Rinard for encouraging me to apply to the PhD program. It is probably fair to say that I would not be earning my PhD without his support. Martin has never stopped challenging me and I will forever aspire to fulfill his oft-repeated directive: "be brilliant!" I would also like to thank Rahul Sarpeshkar for his excellent advice while I was an undergraduate.

I have learned so much from fellow students and postdocs. To begin, I would like to thank Cameron Freer for not only being a fantastic collaborator but also a great friend. Nate Ackerman's infamous "*Nate attack!*" broke us through many an impasse, and his dedication to our joint work and friendship made our long work sessions that much more pleasant. I can only hope that Cameron and Nate learned as much from me as I have from learned from them.

One particular collaboration that inspired many of the questions posed in this dissertation was joint work on the Church language

with Noah Goodman, Vikash Mansinghka, Keith Bonawitz and Josh Tenenbaum. Vikash's world view is dangerously alluring, and his fervent imagination has been a constant source of interesting questions; many ideas in this dissertation can be traced back to late night conversations with Vikash just before and after deadlines. Likewise, Noah Goodman's *probabilistic language of thought* hypothesis has proved fertile ground for fascinating computer science research. My research has benefitted from (and I have thoroughly enjoyed) countless conversations with Chris Baker, Tom Kollar, Timothy O'Donnell, Peter Orbanz, Lorenzo Rosasco, David Sontag, Andreas Stuhlmüller, and many others.

More broadly, I would like to thank all the members of the LIS and CoCoSci groups for their friendship over the years. I am also grateful to the members of the Gatsby Computational Neuroscience Laboratory at UCL for their hospitality during my many visits.

I would like to extend my gratitude to the innumerable teachers and mentors who guided me from Yerba Buena and Lindero Canyon, through Viewpoint and MIT. They will continue to serve as my inspiration in the years to come.

My mother, Meloney, and father, Donald, consistently made tough choices and sacrificed much of their time and no doubt some of their comfort to support my interests and ambitions, and I will remain forever grateful. I am also grateful for the camaraderie of my brothers growing up and to my grandparents, for their love and support.

Finally, to Juliet. Thank you for spending the past ten years by my side.

The work in this dissertation was partially supported by graduate fellowships from the National Science Foundation, MIT's Lincoln Lab, and the Siebel Foundation.

Attribution

The research presented in this dissertation was the product of truly collaborative work that was completed only through the repeated key insights of everyone involved.

Chapters 2 and 3 pertain to computable probability theory and the computability of conditional probability, in particular. This work is the result of collaborations with Nate Ackerman and Cameron Freer, and was originally posted as a preprint [AFR10] on the arXiv in 2010, and then published as an extended abstract [AFR11] in the refereed proceedings of the 26th annual IEEE Symposium on Logic in Computer Science.

Chapter 4 pertains to the computability of exchangeable sequences and their de Finetti measures, and reflects on the problem of computing conditional distributions in this setting. This work is the result of collaborations with Cameron Freer and aspects have been published in various forums. Results on the computability of de Finetti measures were first published as an extended abstract [FR09b] in the refereed proceedings of the 5th Conference on Computability in Europe. A significantly expanded presentation was then posted as a preprint [FR09a] on the arXiv in 2009. Implications for conditional probability in the setting of exchangeable sequences were published as an extended abstract [FR10] appearing in the refereed proceedings of the 13th International Conference on Artificial Intelligence and Statistics.

Chapter 6 presents a new Markov process perspective on Mondrian processes, a novel class of stochastic processes discovered by the author and Yee Whye Teh and first presented at the Neural Information Processing Systems (NIPS) conference in 2008. The results in this dissertation extends those first published in the refereed proceeding of NIPS [RT09]. This work was itself inspired by questions raised during the development of the 'annotated hierarchies' model [RKMT07], joint work with Charles Kemp, Vikash Mansinghka and Joshua Tenenbaum, also published in the proceedings of NIPS in 2007.

The theoretical study of the computability of exchangeability and conditional probability were inspired by practical questions raised during the development of the Church probabilistic programming language [GMR$^+$08], joint work with Noah Goodman, Vikash Mansinghka, Keith Bonawitz, and Joshua Tenenbaum.

The research presented in this dissertation benefitted from the feedback of many colleagues, including Scott Aaronson, Nate Ackerman, Jeremy Avigad, Henry Cohn, Quinn Culver, Cameron Freer, Zoubin Ghahramani, Noah Goodman, Leslie Kaelbling, Charles Kemp, Oleg

Kiselyov, Bjørn Kjos-Hanssen, Vikash Mansinghka, Timothy O'Donnell, Peter Orbanz, Geoff Patterson, Hartley Rogers, Ruslan Salakhutdinov, Chung-chieh Shan, David Sontag, Michael Sipser, Yee Whye Teh, Joshua Tenenbaum, David Wingate, and many anonymous referees.

Contents

Front matter 5
Acknowledgments 6
Attribution 8

Chapter I. Introduction 13
1. Representations of uncertainty 16
2. Outline 21

Chapter II. Computable probability theory 25
1. Basic notions 25
2. Computable metric spaces 28
3. Computable random variables and distributions 31
4. Almost decidable sets 35

Chapter III. Conditional distributions 37
1. Computational limits of probabilistic inference 37
2. Computable conditional distributions 40
3. Computable conditional distributions 43
4. Discontinuous conditional distributions 52
5. Noncomputable almost continuous conditional distributions 54
6. Noncomputable continuous conditional distributions 59
7. Conditioning is Turing jump computable 63
8. Continuity in the setting of identifiability in the limit 64

Chapter IV. Exchangeable sequences and de Finetti's theorem 69
1. de Finetti's Theorem 71
2. Computable Representations 75
3. The Computable Moment Problem 82
4. Proof of the Computable de Finetti Theorem 85
5. Exchangeability in Probabilistic Programs 92
6. Predictive distributions and posterior analysis in exchangeable sequences 100

Chapter V. Distributions on data structures: a case study 107
1. Exchangeable arrays 108
2. Random kd-trees 111

3. Guillotine partitions and Mondrian processes 112
4. Conditional Mondrian processes 125
5. Mondrian processes on unbounded spaces 130

Chapter VI. Conclusion 133

Bibliography 135

CHAPTER I

Introduction

The 1989 *Report of the ACM Task Force on the Core of Computer Science* characterized the discipline of computing as

> the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all computing is 'What can be (efficiently) automated?' [CGM+89]

This dissertation presents a study of algorithmic processes that describe and transform *uncertain* information. In particular, we investigate the class of probability distributions that can be represented *by algorithms* and characterize the fundamental limitations of using this representation to describe and compute conditional distributions.

The motivation for this work comes from the study of *probabilistic programming* and its application to Artificial Intelligence (AI). While the results presented in this dissertation are theoretical in nature and concern the computational limits of distributions in complete generality, the genesis of the ideas and the aesthetics that guided the definitions and theorems were firmly rooted in an effort to significantly extend our ability to represent and reason about complex processes in our uncertain world.

This dissertation studies three important questions at the intersection of computer science, statistics, and probability theory:

The first pertains to **conditional probability**, a cornerstone of Bayesian

statistics and a fundamental notion in probability theory. Kolmogorov's axiomatization of conditional probability [Kol33] was a major advancement at the time, but Kolmogorov's definition gives no recipe for calculation, instead defining conditional probability implicitly by the properties that it must satisfy. Over the course of the last 80 years, there have been a number of proposed constructive definitions of conditional probability, but these approaches have been largely insensitive to computability. Computational issues are paramount in fields like AI and statistics where practitioners are building ever more complex probabilistic models. Computability issues, in particular, are critical to researchers building implementations of probabilistic programming languages with the goal of providing universal Bayesian inference for large classes of stochastic processes.

This work takes a different approach: By looking at conditional probability from a computability-theoretic perspective, we gain new insights into both the mathematical and computational structure of conditional probability. In short, we show that *there is no possible algorithm to calculate conditional probabilities in general*. The result is an analogue in Bayesian statistics of the noncomputability of the halting problem, and can be used to guide extensions of existing probabilistic programming languages to support a much larger class of probabilistic inference problems.

The second question is concerned with **exchangeability**, another fundamental notion in Bayesian statistics and an active area of research in probability theory. Exchangeability underpins a deep relationship between symmetry and the tractability of inference in probabilistic programs. One of the most important theorems pertaining to exchangeability is *de Finetti's theorem*, which states that every exchangeable process can be described as a conditionally independent process. We prove a computable version of de Finetti's theorem and show the surprising fact that it is possible to automatically transform a probabilistic program representing an exchangeable process into a probabilistic program representing a conditionally independent process. These alternative representations have different computational characteristics: the generation of the next element in an exchangeable process will often depend on the state of other elements, forcing the process to proceed in a serial fashion; in contrast, the generation of a conditionally independent process can proceed in parallel. The computable de Finetti theorem also gives us more insight into the computability of conditional probability; using the result, we are able to prove that a much wider class of statistical inference problems is computable than was previously

thought. These results connect the semantics of probabilistic programs with an important result in probability theory, and motivate the study of other types of exchangeability that arise in probabilistic programs.

The third and final question pertains to the problem of constructing **distributions on infinite data structures**, a computational analogue of the mathematical problem of constructing **stochastic processes**. Within Bayesian statistics, the design, study and application of stochastic processes in probabilistic modeling is the purview of Bayesian nonparametrics, and the demonstrated success of this approach is leading to its broader adoption.

Heretofore, a small number of core stochastic processes have served as the building blocks for nonparametric probabilistic models. Probabilistic programming is likely to disrupt this monopoly, as probabilistic programming languages, and especially functional ones, make it much easier to define new and interesting stochastic processes using recursion and abstractions like higher-order procedures, continuations, and data structures (see [GMR+08, RMGT08] for examples).

At the same time, this power makes it easy to construct stochastic processes whose distributions encode nonsensical assumptions about one's uncertainty, and so care must be taken. We present a case study where we construct a stochastic process that satisfies a desired property: in particular, the problem of constructing a model for relational data that satisfies a notion of exchangeability is reduced to the problem of constructing an infinite random partition of a product space. We show that an essential *self-similarity* property is achieved by a recursive algorithm for constructing a random kd-tree data structure. The underlying stochastic process, called a Mondrian process, is connected to active areas of research in probability theory, in particular combinatorial stochastic processes and fragmentation processes.

This dissertation works within the formalism for studying the computability of real numbers introduced by Turing in his foundational paper, "On computable numbers, with an application to the Entscheidungsproblem" [Tur36], and the study of computable real functions and higher-type computability originating in the work by Grzegorczyk [Grz57], Kleene [Kle59], Mazur [Maz63], and others.

More recently, a robust theory of computability for measures and distributions on topological and metric spaces has emerged and several natural proposals have been shown to be equivalent (see, e.g., [AES00] and [SS06]). Roughly speaking, it is possible to exactly sample a value from a distribution on some topological space S if and only if it is

possible to compute, to arbitrary accuracy, the measures $\mu(A_i)$ of some countable basis $\{A_i\}$ of open sets in S. The latter property has been used to develop a theory for computable integration. We build on this framework to study important operations and objects in probability theory.

The study of computable distributions has implications beyond theoretical computer science. Our theoretical results pertain to the probabilistic programming approach to AI, and so should be of interest to AI practitioners as well. By studying computational limits in this abstract setting, we can identify constraints that hold for all probabilistic programming languages. For example, the noncomputability of conditional probability implies that any implementation of a sufficiently powerful probabilistic programming language will necessarily have to choose a number of special cases of conditioning to support, as no algorithm exists that supports all cases.

More broadly, the increasing role of uncertainty in everyday computing and decision-making is drawing computer science, probability and statistics into ever closer contact. The work presented in this dissertation takes us a step toward a theory of probabilistic computation suitable for guiding theoreticians and practitioners alike in the design of systems that can scale to meet the statistical needs of our increasingly complex data-driven world.

1. Representations of uncertainty

The flexible and efficient representation of uncertainty is now considered a central problem of AI. The emphasis on deductive, rule-based reasoning that characterized the so-called *good old fashioned AI* gave way after the realization that inductive reasoning, and in particular probabilistic inference, was far more useful for building adaptive systems with access to only partial information. More recently, it has become clear that in order to model complex phenomena like vision, communication and planning, we need a representation of uncertainty that is powerful enough to capture very general stochastic processes. In particular, the graphical model formalism that ushered in an era of rapid progress in AI has proven inadequate in the face of these new challenges. A promising new approach that aims to bridge this gap is probabilistic programming, which marries probability theory, statistics and programming languages.

Probability theory provides the necessary mathematical formalism for representing uncertainty and incorporating new evidence. In particular, beliefs about a set H of hypotheses and set E of possible

observations are encoded as a probability distribution over the product space $H \times E$, and observed evidence is incorporated by restricting (and renormalizing) the distribution to the subspace compatible with the actual observations or, in other words, by forming the *conditional* distribution. However, probability theory remains silent on the question of what space of hypotheses one should consider when building a probabilistic model, how best to specify a distribution on that space in order to reflect one's uncertainty, and how to efficiently update the distribution in light of new observations.

Putting aside the question of what space of hypotheses one should consider, even the task of specifying a probabilistic model over a finite number of binary random variables is nontrivial. Naively, before any statistical inferences can take place, a probability must be assigned to every possible configuration of the random variables. For n binary variables, there are 2^n such configurations, ruling out an exhaustive tabular representation for all but the smallest models. Furthermore, such a representation for a distribution can make important operations very expensive to compute; e.g., calculating a marginal or conditional probability may require the summation of an exponential number of entries.

A major advancement in the search for a representation of uncertainty supporting efficient inference was the introduction of the graphical model formalism (see [Pea88] for more details and a historical summary). A key insight in the development of graphical models was the identification of conditional independence structure as useful both for the compact specification of a probabilistic model and for the efficient computation of marginal and conditional probabilities.

Given a distribution over a (finitely or countably) indexed collection $\{X_v\}_{v \in V}$ of random variables, a *directed graphical model* (or *Bayesian network*) is a directed acyclic graph (V, E) where the vertices V index the random variables and the edges $E \subseteq V \times V$ encode the conditional independence structure. In particular, let $\mathrm{Pa}(v) = \{u : (u, v) \in E\}$ denote the set of parent vertices of v. Then there exist independent uniform random variables $\{\xi_v\}_{v \in V}$ such that, for all vertices $v \in V$,

$$X_v = g_v(\xi_v, \{X_u\}_{u \in \mathrm{Pa}(v)}) \qquad \text{a.s.} \tag{1}$$

for some measurable function g_v.

The "local relationship" g_v is uniquely characterized (up to a measure preserving transformation) by the conditional distribution of X_v given its parents $\{X_u\}_{u \in \mathrm{Pa}(v)}$. Returning to the case of binary random variables, the specification of an arbitrary distribution whose conditional independence structure satisfies a directed graphical model (V, E)

would require that we specify only $\sum_{v \in V} 2^{|\mathrm{Pa}(v)|}$ conditional probabilities. Clearly the savings are considerable for many graphs.

The graph structure is also crucial for developing efficient inference. Given any set $C \subseteq V$ of conditioned variables, a graph-theoretic condition called *d-separation* enables one to efficiently decide whether any particular conditional independence is implied by the graph (see [Pea88, §3.3.1] for more details). As Jordan [Jor10] has recently argued, the marriage of graph theory and probability theory has enabled deep results from combinatorics to be brought to bear on the problem of efficiently evaluating the exponential sums and products that characterize marginal and conditional probabilities. More broadly, concrete representational choices have focused research and led to rapid scientific progress on the relationship between a representation of a probabilistic model and the efficiency with which statistical inference can be performed.

However, just as probability theory is agnostic to the semantics and concrete representation of a probability space under consideration, the theory of graphical models is (relatively) silent on the question of how to concretely represent the graph (V, E) and the local relationships $g_v(\xi_v, \cdot)$ or, equivalently, the local conditional distributions.

A common concrete representation of a graphical model is a finite list of vertices and edges, along with a specification for a parametric form for the local conditional distributions. Clearly such a representation restricts our attention to finite graphical models. And while most models can be approximated arbitrarily well by a finite model of some dimension, such an approach is likely to obscure the simplicity of some processes. Just as it may be possible to write a complex computer program as a long list of primitive instructions, expressing a graphical model as a flattened list of its edges obscures any abstract patterns that aided in its design or discovery or that could aid in its generalization or implementation.

The pun between a graphical model and its presentation as a literal graphic has been a significant psychological impediment to the effective use of graphical models in more general settings. Other authors have recognized this problem and have proposed new representations and alternative formalisms. For example, Buntine [Bun94] introduced *plate notation*, which can be used to indicate that a subgraph is replicated, much like a `for` loop allows repetition in a program. Other formalisms, such as Probabilistic Relational Models [KP97, FKP99] and Markov Logic Networks [RD06] go further and enable families of large finite

graphical models to be expressed compactly in terms of a finite set of relational expressions.

New trends in Bayesian nonparametric statistics stress the limits of these representations. In this setting, classical finite-dimensional prior distributions are replaced by stochastic processes, i.e., indexed collections of (possibly infinitely many) random variables. Using stochastic processes, it is possible to put distributions on infinite structures like functions, graphs, or even distributions. While popular stochastic processes often have compact mathematical descriptions, graphical models can only represent the high-level structure of these models. As a result, inference algorithms that use graphical representations cannot be applied to this new class of models and, often, it is necessary to create special purpose inference algorithms for each new model.

However, the most challenging models from a representational standpoint are those of phenomena such as grammatical structure in language, multi-party communication, and optimal planning as inference. In each of these cases, the static conditional independence structure corresponds to an extremely large (if not infinite) graphical model.

1.1. Probabilistic programs.

> But the fundamental reason for [the] inadequacy of traditional grammars is a more technical one. Although it was well understood that linguistic processes are in some sense "creative," the technical devices for expressing a system of recursive processes were simply not available until much more recently. In fact, a real understanding of how a language can (in Humboldt's words) "make infinite use of finite means" has developed only within the last thirty years, in the course of studies in the foundations of mathematics.
>
> Noam Chomsky
> *Aspects of the Theory of Syntax*, 1969.

In the similar way, the inadequacy of probabilistic accounts of complex phenomena is a technical one: Graphical models capture the static conditional independence structure of a distribution. However, phenomena like communication and planning are fundamentally dynamic, and their adequate representation requires technical devices for expressing recursive processes.

Probabilistic programs are such a device.

Probabilistic programming languages have been the subject of intense research since the 1950s in subfields of computer science including programming languages (e.g., [SD78, MMS96]), domain theory (e.g., [JP89, Esc09]) and formal methods (e.g., [MM99, Hur02, HT07]). However, their use in statistical AI began much more recently with the introduction of Pfeffer's IBAL language [Pfe01], and since then a number of probabilistic functional programming languages (e.g, $\lambda_\circ$[PPT08], Church [GMR$^+$08], and HANSEI [KS09]) have been proposed within the AI community.

Many probabilistic programming languages extend existing deterministic programming languages. As a result, they inherit modern programming language features that were introduced to help mitigate the complexity of designing large software systems. These same features can be brought to bear on the task of designing complex probabilistic models. As a result of powerful means of combination and abstraction, such languages, especially those built on functional programming languages, can naturally represent the higher-order objects used in statistical AI and machine learning (e.g., a distribution on graphs, or a distribution on distributions).

Probabilistic programs can also support efficient inference. The syntactical structure of a probabilistic programs and its runtime call graph expose fine-grained conditional independence structure that be

used to implement a wide range of approaches to inference, including, e.g., exact lazy enumeration using continuations [Rad07], importance sampling [PPT08], variational message passing [MW08], and Markov Chain Monte Carlo sampling [GMR$^+$08].

1.2. Limits of probabilistic programming. As exciting as this new approach is, the application to statistical problems is relatively new and deserves study. This dissertation explores the theoretical limits and potential of the probabilistic programming languages approach to statistical AI.

A fundamental characteristic of any adaptive system is the ability to update its behavior in light of new observations and data. In a Bayesian statistical setting, this is achieved by computing conditional distributions, and as a result, almost all probabilistic programming languages support conditioning to some degree of generality, although this degree varies substantially. In particular, every existing language provides incomplete and ad hoc support for conditioning continuous random variables. This dissertation explains why such support is necessarily incomplete.

The probabilistic programming approach has the potential to revolutionize not only statistical modeling, but also statistical computation. The adoption of the graphical model formalism led to rapid progress in the theoretical understanding of the complexity of inference and the development of efficient inference algorithms; probabilistic programming languages extend this research program and offer a challenging yet tantalizing new target for researchers. However, this generality comes at a price: we show that there are joint distributions represented as probabilistic programs whose conditional distributions have no such representation. Therefore, in order to build a research program that can focus effort in the way that the graphical model formalism did, we must first decide how to *define* the canonical inference problem for probabilistic program inference. In order to do this, we must characterize when we can compute conditional distributions.

2. Outline

In Chapter II, we introduce existing notions of computability on topological and metric spaces, and then focus on probability measures on their Borel σ-algebras. A probabilistic program representation of a distribution is formalized as a computable random variable from a basic probability space (which represents a source of randomness). We argue that the appropriate notion of computability of random variables (and

measurable functions more generally) requires that they be continuous outside of a null set.

We then turn to a study of the computability of conditional probability, a fundamental probabilistic operation of critical importance in statistical applications. We ask:

> "If there is an algorithm for a joint distribution on a pair (X, Y) of random variables (e.g., representing our *a priori* beliefs in a Bayesian setting), is there an algorithm for the conditional distribution $\mathbf{P}[\mathsf{X}|\mathsf{Y} = y]$ (representing our *a posteriori* beliefs given an observation)?"

This question arose naturally from work on the probabilistic programming language Church [GMR$^+$08]. One of its implementations, MIT-Church, is an algorithm for producing samples from the conditional distributions of uncertain (i.e., random) variables given observed data. What are the inherent limits of this endeavor? How general can such algorithms be?

We answer these questions by constructing a pair of computable continuous random variables whose conditional distribution is nonetheless not computable. This, in turn, implies that there is no algorithm for computing conditional probabilities or expectations in the general case, thereby explaining why existing probabilistic programming language systems necessarily have incomplete support for conditioning on continuous random variables.

In order to begin to square these results with the fact that conditional distributions are computed regularly in practice, we identify additional structure which renders conditioning computable. For example, under mild additional assumptions, conditioning on discrete random variables or general random variables with bounded joint densities is computable. We highlight a surprising connection with information theory, whereby suitably smooth computable noise on the measurement ensures the computability of the conditional distribution.

In Chapter IV, we turn to the setting of exchangeable sequences of random variables, where de Finetti's classic representation theorem tells us that there always exists a so-called directing random measure that renders the sequence conditionally i.i.d. In this setting, conditional densities may not exist, yet computational practice suggests that the conditional distributions are often computable. We ask:

> "If there is an algorithm for an exchangeable sequence, is there an algorithm for its de Finetti measure?"

This question strikes at the heart of what appears to be a natural affinity between higher-order probabilistic programs and Bayesian nonparametric statistics. The motivating example was that of the Indian Buffet Process and its de Finetti measure, the beta process. Existing algorithms [WI98, TGG07] provide only *approximate* samples from the latter process, and there is no known exact algorithm. We prove a computable version of de Finetti's theorem and then use it to give necessary and sufficient conditions for the computability of conditional distributions involving exchangeable sequences and their directing random measures.

In Chapter V, we study the problem of constructing a stochastic process using a recursive algorithm. We present a novel stochastic process with a self-similarity property that we use to construct an infinite kd-tree data structure.

We now turn our attention to the fundamental notions of computability for distributions and random variables.

CHAPTER II

Computable probability theory

Computable probability theory is a framework for studying probabilistic operations as performed by algorithms. The theory is based on a long tradition of mathematical work in recursion theory (see, e.g., [Tur36, Kle59]) and computable analysis (for a survey, see, e.g., [Wei00a]) studying the computability of reals, continuous functions and higher types, and also builds upon work in domain theory and the semantics of programming languages (see, e.g., [Eda97]). In this chapter, we present existing notions from computability probability theory, but also introduce the notion of a *computable random variable*, which we take to be a formalization of the notion of a probabilistic program. We begin with a review of classical computability theory, and then introduce more recent notions of computability on metric spaces. In chapters III and IV, we present new results in computability probability theory.

In examining the computability of probabilistic operations, we are concerned with tasks that we can perform, sometimes to arbitrary accuracy, and also with what we cannot do, even approximately. Some results in computable probability theory, such as our computable extension of de Finetti's theorem (described in Chapter IV) provide explicit algorithms. Other results, such as the noncomputability of conditioning (described in Chapter III) prove the fundamental nonexistence of algorithms to perform certain tasks.

In situations where there is provably no exact algorithm to perform an operation, it is sometimes possible to improve such results, using techniques from computability theory, to show the impossibility of always computing non-trivial approximations, let alone arbitrarily good ones. Hence computable probability is not just about the possibilities and limitations of exact computation, but is also directly relevant to floating point and fixed precision calculations.

1. Basic notions

Objects like real numbers and probability measures have, in general, only infinite descriptions. In contrast, algorithms have finite descriptions,

and actual computers can only perform a finite number of operations in any finite amount of time, although an algorithm may run for an unbounded amount of time. We are therefore especially interested in those particular reals, probability measures, and other infinite structures that admit finite descriptions. We begin by recalling some elementary definitions from computability theory, which will form the foundation for suitable theories on more abstract spaces. (For more details on recursion theory, see Rogers [Rog87, Ch. 5].)

We say that a partial function $f : \mathbb{N} \to \mathbb{N}$ is **partial computable** when there is some Turing machine that, on input $n \in \operatorname{dom}(f)$, eventually outputs $f(n)$ on its output tape and halts, and on input $n \notin \operatorname{dom}(f)$, never halts. We say that a function $f : \mathbb{N} \to \mathbb{N}$ is **total computable** or simply **computable** when it is a partial computable function that is total, i.e., $\operatorname{dom}(f) = \mathbb{N}$.

Using this definition, we can define computability on other structures. For example, a sequence $\{a_0, a_1, \dots\}$ of elements in $\mathbb{N}$, or equivalently, a total function from $\mathbb{N}$ to $\mathbb{N}$, is computable when it is total computable when viewed as a function. Finite sequences (i.e., tuples) can likewise be viewed as computable when viewed as partial computable functions.

For an arbitrary partial function g, let $g(x)\!\downarrow$ denote that g is defined at x and let $g(x)\!\uparrow$ denote that g is undefined at x. A fundamental result in computability theory is that there exists an enumeration $\varphi_1, \varphi_2, \dots$ of the partial computable functions such that the partial function

$$\operatorname{apply}(e, x) = \begin{cases} \varphi_e(x), & \text{if } \varphi_e(x)\!\downarrow \\ \text{undefined}, & \text{if } \varphi_e(x)\!\uparrow \end{cases} \tag{2}$$

is computable, where $\varphi_e(x)$ is the value that the eth partial function outputs on input x, if defined.

A simple counting argument shows that there must be noncomputable functions, as there are uncountably many functions from $\mathbb{N}$ to $\mathbb{N}$ and only countably many Turing machines. Turing's diagonal argument showing that the halting problem is undecidable implies that the function

$$h(e) = \begin{cases} 1, & \text{if } \varphi_e(0)\!\downarrow \\ 0, & \text{if } \varphi_e(0)\!\uparrow \end{cases} \tag{3}$$

is not computable [Tur36].

Having defined computability on the function space $\mathbb{N}^{\mathbb{N}}$, we now study notions of computability on $2^{\mathbb{N}}$, i.e., subsets of $\mathbb{N}$. A subset $D \subseteq \mathbb{N}$

is said to be **computable** when its characteristic function

$$\chi_D(n) = \begin{cases} 1, & \text{if } n \in D \\ 0, & \text{if } n \notin D \end{cases} \tag{4}$$

is computable. The noncomputability of the function h above shows that the domain of a partial computable function is not necessarily a computable subset. A set is said to be **computably enumerable** (or c.e.) when it is the domain of a partial computable function. A set is said to be co-c.e. when its complement is c.e. It is easy to show that a set is computable if and only if it is both c.e. and co-c.e, and that a set is c.e. iff it is empty or the range of a total computable function (thereby justifying the term *enumerable*).

These notions form the foundation of computability on richer spaces. The above development will be seen to replay itself in more abstract settings.

To begin, let S and T be countable sets. In order to construct a notion of computability on T^S and 2^S, we fix an enumeration $s_0, s_1, s_2, \ldots$ of S and $t_0, t_1, t_2, \ldots$ of T and then consider the computability of the corresponding elements in $\mathbb{N}^{\mathbb{N}}$ and $2^{\mathbb{N}}$. For example, a partial function $f : S \to T$ is said to be **partial computable *(relative to the enumerations s of S and t of T)*** when there is a partial computable function $g : \mathbb{N} \to \mathbb{N}$ such that $t_{g(n)} = f(s_n)$ for every $n \in \mathbb{N}$ such that $s_n \in \mathrm{dom}(f)$. As before, a subset $D \subseteq S$ is said to be computable when its characteristic function $\chi_D : S \to \{0, 1\}$ is computable (relative to the enumeration of S). Likewise, D is c.e. (relative to the enumeration of S) when it is the domain of a partial computable function and co-c.e. (relative to the enumeration of S) when its complement is c.e. (relative to the enumeration of S). We will elide the mention of the enumeration when it is clear from context.

Example II.1. A few enumerations are important enough going forward that we pause briefly to highlight them.

(1) For all $k \in \mathbb{N}$, there is an enumeration t of $\mathbb{N}^k$ such that, for each $i \in \{1, \ldots, k\}$, the projection $\pi_i : \mathbb{N}^k \to \mathbb{N}$ given by $\pi_i(n_1, \ldots, n_k) = n_i$ is computable relative to t. In particular, consider the case where $k = 2$. The Cantor pairing function $\langle \cdot, \cdot \rangle : \mathbb{N}^2 \to \mathbb{N}$ given by

$$\langle a, b \rangle = \frac{1}{2}(a + b)(a + b + 1) + b, \tag{5}$$

bijectively maps pairs of natural numbers to natural numbers and renders the projections computable. Iterating the pairing

function enables us to encode tuples of higher (or arbitrary) arity.

(2) There exists an enumeration of the rationals $\mathbb{Q}$ with respect to which the embedding $\mathbb{N} \hookrightarrow \mathbb{Q}$, as well as addition, subtraction, multiplication, division and most other natural operations are computable. Concretely, we can represent rationals as tuples in terms of their representation as (signed) fractions of natural numbers in reduced form.

Remark II.2. In this chapter and beyond, we will often elide the enumerations at play when it is straightforward to choose enumerations that ensure the desired primitive operations are computable. E.g., the space of finite sequences of integers can easily be represented so as to allow one to compute the length of such a sequence, extract elements at particular coordinates, etc.

We now proceed beyond finite and countable sets.

2. Computable metric spaces

In this section, we present basic notions and results for computable metric spaces. The origins of this theory can be traced back to the study of the computability of real functions and functionals initiated by Grzegorczyk [Grz57], Kleene [Kle59], Mazur [Maz63], and others.

More recently, Pour-El and Richards [PER89], Weihrauch [Wei89], and others have brought in methods from constructive analysis. There has been much recent work following their approach to computable analysis, often referred to as the Type-2 Theory of Effectivity (TTE). In this approach, one typically asks for a continuous real function to have a computable modulus of continuity. This leads to a robust theory, which lines up with exact real arithmetic on computers. Furthermore, with computable continuity, one can perform something like automatic numerical analysis. These notions extend to more general spaces.

Computable metric spaces, as developed in computable analysis, provide a convenient and general framework for formulating results in computable probability theory. For consistency, we largely use definitions from [HR09] and [GHR10]; imported definitions and results are clearly cited. Additional details about computable metric spaces can also be found in [Wei00a, Ch. 8.1] and [Gác05, §B.3], and their relationship to computable topological spaces is explored in [GSW07]. Computable measures on metric spaces have also been studied using domain theory [EH98].

We first recall basic notions of computability for real numbers, which extend back to Turing's foundational paper [Tur36] (see also [Wei00a,

Ch. 4.2] and [Nie09, Ch. 1.8]). We say that a real r is a *c.e. real* when the set of rationals $\{q \in \mathbb{Q} : q < r\}$ is c.e. Similarly, a *co-c.e. real* is one for which $\{q \in \mathbb{Q} : q > r\}$ is c.e. (C.e. reals are sometimes called *left-c.e.* or *lower-semicomputable* reals, while co-c.e. reals are sometimes called *right-c.e.* or *upper-semicomputable* reals.) A real r is *computable* when it is both c.e. and co-c.e. Equivalently, a real r is computable when there is a computable sequence of rationals $\{q_k\}_{k\in\mathbb{N}}$ such that $|q_k - r| < 2^{-k}$ for all $k \in \mathbb{N}$.

We can now define more abstract notions:

Definition II.3 (Computable metric space [GHR10, Def. 2.3.1])**.** A **computable metric space** is a triple $(S, \delta, \mathcal{D})$ for which δ is a metric on the set S satisfying

(1) (S, δ) is a complete separable metric space;
(2) $\mathcal{D} = \{s_i\}_{i\in\mathbb{N}}$ is an enumeration of a dense subset of S, called **ideal points**; and,
(3) $\delta(s_i, s_j)$ is computable, *uniformly* in i and j; i.e., there is a total computable function $f : \mathbb{N}^3 \to \mathbb{Q}$ such that
$$|f(i, j, k) - \delta(s_i, s_j)| < 2^{-k}.$$

For a point $s \in S$ and positive real r, let $B(s, r)$ denote the radius-r open ball centered at s. We call the set

$$\mathscr{B}_S := \{B(s_i, q_j) \,:\, s_i \in \mathcal{D},\ q_j \in \mathbb{Q},\ q_j > 0\} \tag{6}$$

the **ideal balls of** S, and fix the canonical enumeration induced by pairing the enumerations of $\mathcal{D}$ and $\mathbb{Q}$.

Example II.4.
(1) The two-point set $\{0, 1\}$ is a computable metric space under the discrete metric, given by $\delta(0, 1) = 1$. A similar approach can be used to make any countable set (like $\mathbb{N}$) into a computable metric space once we fix an enumeration. In the case of $\mathbb{N}$, the elementary notions agree with those that can be derived by treating the space as a computable metric space.
(2) Cantor space is the set $\{0, 1\}^{\mathbb{N}}$ of infinite binary sequences under the metric $\delta_{\{0,1\}^{\mathbb{N}}}(x, y) = 2^{-k}$, where k is the index of first term on which the sequences x and y differ. (The induced topology is the therefore the product topology.) The set of eventually constant sequences is dense, and (under, e.g., the enumeration induced by the standard enumeration of finite strings) makes $\{0, 1\}^{\mathbb{N}}$ into a computable metric space.
(3) The set $\mathbb{R}$ of real numbers is a metric space under the Euclidean metric. The set $\mathbb{Q}$ of rationals is dense in $\mathbb{R}$ and (under its

standard enumeration) makes $\mathbb{R}$ into a computable metric space. The same can be shown of $\mathbb{R}^n$, for $n \in \mathbb{N}$, and the space $\mathbb{R}^\infty$ of infinite sequences of reals.

Definition II.5 (Computable point [GHR10, Def. 2.3.2])**.** Let $(S, \delta, \mathcal{D})$ be a computable metric space. A point $x \in S$ is **computable** when there is a computable sequence $x_0, x_1, \ldots$ in $\mathcal{D}$ such that $d(x_n, x) < 2^{-n}$ for all $n \in \mathbb{N}$. We call such a sequence $\{x_n\}_{n\in\mathbb{N}}$ a **representation** of the point x.

Remark II.6. A real $\alpha \in \mathbb{R}$ is computable (as in Section 2) if and only if α is a computable point of $\mathbb{R}$ (as a computable metric space). Although most of the familiar reals are computable, there are only countably many computable reals, and therefore, there are uncountably many noncomputable reals.

Many well-known noncomputable reals, such as the *halting probability* Ω (sometimes called *Chaitin's constant* [Cha75]), can be constructed in terms of universal Turing machines or by explicit diagonalizations. But noncomputable reals can also arise via general constructions; for example, the limit of a computable sequence of computable reals is typically not even c.e. or co-c.e. (for more details, see, e.g., [Zhe02, §9]).

We now consider the computability of *sets* of reals. The notion of a c.e. open set is fundamental in classical computability theory, and admits a simple definition in an arbitrary computable metric space.

Definition II.7 (C.e. open set [GHR10, Def. 2.3.3])**.** Let $(S, \delta, \mathcal{D})$ be a computable metric space with the corresponding enumeration $\{B_i\}_{i\in\mathbb{N}}$ of the ideal open balls $\mathscr{B}_S$. We say that $U \subseteq S$ is a **c.e. open set** when there is some c.e. set $E \subseteq \mathbb{N}$ such that $U = \bigcup_{i\in E} B_i$.

(The c.e. open sets can also be seen as the computable points in a suitable computable *topological* space of sets, or equivalently, as those sets whose characteristic functions are *lower semicomputable*. We discuss notions of computability on topological spaces in Chapter IV.)

Note that the class of c.e. open sets is closed under computable unions and finite intersections.

A computable function can be thought of as a continuous function whose local modulus of continuity is computable. It is important to consider the computability of *partial* functions, since many natural and important random variables are continuous only on a measure one subset of their domain.

Definition II.8 (Computable partial function [GHR10, Def. 2.3.6])**.** Let $(S, \delta_S, \mathcal{D}_S)$ and $(T, \delta_T, \mathcal{D}_T)$ be computable metric spaces, the latter

with the corresponding enumeration $\{B_n\}_{n\in\mathbb{N}}$ of the ideal open balls $\mathscr{B}_T$. A function $f : S \to T$ is said to be **computable on** $R \subseteq S$ when there is a computable sequence $\{U_n\}_{n\in\mathbb{N}}$ of c.e. open sets $U_n \subseteq S$ such that $f^{-1}(B_n) \cap R = U_n \cap R$ for all $n \in \mathbb{N}$.

Note that we will sometimes forego explicitly listing the ideal points and associated metric.

Remark II.9. Let S and T be computable metric spaces. Then $f : S \to T$ is computable on all of S if and only if the inverse image $f^{-1}(V)$ of a c.e. open set V is itself a c.e. open set, uniformly in V.

Remark II.10. Let S and T be computable metric spaces. If $f : S \to T$ is computable on some subset $R \subseteq S$, then for every *computable* point $x \in R$, the point $f(x)$ is also computable. One can show that f is computable on R when there is a program that uniformly transforms representations of points in R to representations of points in S. (For more details, see [HR09, Prop. 3.3.2].)

3. Computable random variables and distributions

A relatively recent synthesis of results from computable analysis and effective domain theory has characterized computable measures on a wide variety of topological and metric spaces. However, the computability theory of probability distributions extends back to work of de Leeuw, Moore, Shannon, and Shapiro [dMSS56].

Within the framework of computable analysis, computable probability measures have been analyzed by Weihrauch [Wei99], Schröder [Sch07], and others. Another approach to computable probability comes from domain theory and an analysis of how computations on continuous structures are formed from partial information on an appropriate partial order. Early work is due to Scott [Sco75], Plotkin [Plo76], and many others, and more recent work on representing probability measures is due to Edalat [Eda96] and others.

Recently, these two threads have converged on essentially equivalent definitions of computable probability measures in a wide range of settings (see, e.g., [AES00] and [SS06]). In the following section, we present this definition of a computable probability measure on a computable metric space, which we will use in Chapter III as the basis of our study of the computability of conditional probability. In Chapter IV, we present our results using compatible notions for computable topological spaces.

Before we characterize the computable points in the space of Borel probability measures on a computable metric space, we introduce the

notion of a *computable random variable*, as it will suggest an appropriate metric to use for defining computable measures.

Intuitively, a random variable maps an input source of randomness to an output, inducing a distribution on the output space. Here we will use a sequence of independent fair coin flips as our source of randomness. This generates the same rich class of computable distributions as that generated by more sophisticated sources, such as i.i.d. uniform random variables. We formalize the notion of independent fair coin flips via a probability measure $\mathbf{P}$ on the space $\{0,1\}^{\mathbb{N}}$ of infinite binary sequences, defined to be the product measure of the uniform distribution on $\{0,1\}$. (For a detailed explicit construction of $\mathbf{P}$, see [Str05, §6.2.1].)

Henceforth we will take $(\{0,1\}^{\mathbb{N}}, \mathbf{P})$ to be the basic probability space, unless otherwise stated. (We will later see that this choice is not especially restrictive.) We will typically use a sans serif font for random variables.

Definition II.11 (Random variable and its pushforward)**.** Let S be a computable metric space. A **random variable in** S is a function $\mathsf{X} : \{0,1\}^{\mathbb{N}} \to S$ that is measurable with respect to the Borel σ-algebras of $\{0,1\}^{\mathbb{N}}$ and S. We will denote by $\mathbf{P}_{\mathsf{X}}$ the probability measure on S given by

$$\mathbf{P}_{\mathsf{X}}(B) := (\mathbf{P} \circ \mathsf{X}^{-1})(B) = \mathbf{P}(\mathsf{X}^{-1}(B)) = \mathbf{P}\{\mathsf{X} \in B\}, \tag{7}$$

for Borel sets $B \subseteq S$. We say that $\mathbf{P}_{\mathsf{X}}$ is the **distribution of** X.

Definition II.12 (Computable random variable)**.** Let S be a computable metric space. Then a random variable X in S is a **computable random variable** when X is computable on some $\mathbf{P}$-measure one subset of $\{0,1\}^{\mathbb{N}}$.

(This definition is closely related to that of an *almost computable* function [HR09, Def. 5.2] and of a *computable almost everywhere* function [Bos08, Def. 3.4]. Another approach to computable random variables is given in [Sch07, §3.3].)

Remark II.13. Let $(S, \delta_S, \mathcal{D}_S)$ be a computable metric space. Intuitively, X is a computable random variable when there is a program that, given access to an oracle bit tape $\omega \in \{0,1\}^{\mathbb{N}}$ representing an infinite sequence of random bits, outputs a representation of the point $\mathsf{X}(\omega)$ (i.e., enumerates a sequence $\{x_i\}$ in $\mathcal{D}$ where $\delta(x_i, \mathsf{X}(\omega)) < 2^{-i}$ for all i), for all but a measure zero subset of bit tapes $\omega \in \{0,1\}^{\mathbb{N}}$ (see Remark II.10).

One of the essential properties of a computable random variable is that for every finite portion of the output stream, the program has

consumed only a finite number of input bits. When a random variable does not produce a valid output representation, this means that after some finite number of steps, it then consumes its entire remaining input stream without producing another output.

Even though the source of randomness is a sequence of discrete bits, there are computable random variables with *continuous* distributions, as we now demonstrate by constructing a uniform random variable.

Example II.14 ([FR10, Ex. 3])**.** Given a bit tape $\omega \in \{0,1\}^{\mathbb{N}}$, for each $k \in \mathbb{N}$ define $\mathsf{X}_k(\omega) := \sum_{i=1}^{k} \omega_i 2^{-i}$. Note that, for every ω, we have $|\mathsf{X}_k(\omega) - \mathsf{X}_{k+1}(\omega)| \leq 2^{-(k+1)}$, and so $\mathsf{X} := \lim_k \mathsf{X}_k(\omega)$ exists. Thus the sequence of rationals $\{\mathsf{X}_k(\omega)\}_{k\in\mathbb{N}}$ is a representation of the real number $\mathsf{X}(\omega)$. Furthermore, because each rational $\mathsf{X}_k(\omega)$ is computed using only finitely many bits of ω, the random variable X is computable. It is straightforward to check that the distribution of X (as $\omega \in \{0,1\}^{\mathbb{N}}$ varies according to $\mathbf{P}$) is uniform on $[0,1]$.

This construction can be extended to define a computable i.i.d.-uniform sequence, by splitting up each of the given elements of $\{0,1\}^{\mathbb{N}}$ into countably many disjoint subsequences and dovetailing the constructions [FR10, Ex. 4].

It is crucial that we consider random variables that are required to be computable only on a $\mathbf{P}$-measure one subset of $\{0,1\}^{\mathbb{N}}$. To understand why, consider the following example. For a real $\alpha \in [0,1]$, we say that a binary random variable $\mathsf{X} : \{0,1\}^{\mathbb{N}} \to \{0,1\}$ is a **Bernoulli**(α) random variable when $\mathbf{P}_{\mathsf{X}}\{1\} = \alpha$. There is a Bernoulli$(\frac{1}{2})$ random variable that is computable on all of $\{0,1\}^{\mathbb{N}}$, given by the program that simply outputs the real number corresponding to the first bit of the input sequence. Likewise, when α is **dyadic** (i.e., a rational with denominator a power of 2), there is a Bernoulli(α) random variable that is computable on all of $\{0,1\}^{\mathbb{N}}$. However, this is not possible for any other choices of α (e.g., $\frac{1}{3}$).

Proposition II.15. *Let $\alpha \in [0,1]$ be a nondyadic real number. Every* Bernoulli(α) *random variable $\mathsf{X} : \{0,1\}^{\mathbb{N}} \to \{0,1\}$ is discontinuous, hence not computable on all of $\{0,1\}^{\mathbb{N}}$.*

Proof. Assume X is continuous. Let $Z_0 := \mathsf{X}^{-1}(0)$ and $Z_1 := \mathsf{X}^{-1}(1)$. Then $\{0,1\}^{\mathbb{N}} = Z_0 \cup Z_1$, and so both are closed (as well as open). The compactness of $\{0,1\}^{\mathbb{N}}$ implies that these closed subspaces are also compact, and so Z_0 and Z_1 can each be written as the finite disjoint union of clopen basis elements. But each of these elements has dyadic measure, hence their sum cannot be either α or $1-\alpha$, contradicting the fact that $\mathbf{P}(Z_1) = 1 - \mathbf{P}(Z_0) = \alpha$. □

On the other hand, for an arbitrary computable $\alpha \in [0,1]$, a more sophisticated construction [Man73] produces a Bernoulli(α) random variable that is computable on every point of $\{0,1\}^{\mathbb{N}}$ other than the binary expansion of α. These random variables are manifestly computable in an intuitive sense (and can even be shown to be optimal in their use of input bits, via the classic analysis of rational-weight coins by Knuth and Yao [KY76]). Moreover, they can in fact be computed on real computers using this algorithm. Hence it is natural to admit as computable random variables those measurable functions that are computable only on a $\mathbf{P}$-measure one subset of $\{0,1\}^{\mathbb{N}}$, as we have done.

The notion of a computable measure follows naturally from an analysis of computable random variables. Let X be a computable random variable on a computable metric space $(S, \delta_S, \mathcal{D}_S)$. Consider what we can learn about its distribution $\mathbf{P}_{\mathsf{X}}$ from observing the behavior of X. Note that if a program computing X (on a measure one subset) outputs an integer i encoding an ideal ball B_i, having read only the first k bits $\omega_1 \cdots \omega_k$ of its bit tape, then $\mathbf{P}_{\mathsf{X}}(B_j) \geq 2^{-k}$ for every ideal ball $B_j \supseteq B_i$, because for every bit tape beginning with $\omega_1 \cdots \omega_k$, the program also outputs i. Therefore, given such a program, we can record, for every ideal ball, those finite bit tape prefixes that are mapped to subballs, thereby tabulating arbitrarily good rational lower bounds on its measure. In fact, in a similar manner, we can collect arbitrarily good rational lower bounds on the the measure of an arbitrary finite union of ideal balls. This information classically determines the probability measure $\mathbf{P}_{\mathsf{X}}$. Moreover, it computably determines this measure, as we will see.

Let $\mathcal{M}_1(S)$ denote the set of (Borel) probability measures on a computable metric space S. The Prokhorov metric (and a suitably chosen dense set of measures [Gác05, §B.6.2]) makes $\mathcal{M}_1(S)$ into a computable metric space [HR09, Prop. 4.1.1].

Theorem II.16 ([HR09, Thm. 4.2.1])**.** *Let $(S, \delta_S, \mathcal{D}_S)$ be a computable metric space. A probability measure $\mu \in \mathcal{M}_1(S)$ is computable point of $\mathcal{M}_1(S)$ (under the Prokhorov metric) if and only if the measure $\mu(A)$ of a c.e. open set $A \subseteq S$ is a c.e. real, uniformly (in the index of) A.*

Thus the above analysis of what we learn by simulating a computable random variable X in S shows that $\mathbf{P}_{\mathsf{X}}$ is a computable point in the computable metric space $\mathcal{M}_1(S)$.

Proposition II.17 (Computable random variables have computable distributions [GHR10, Prop. 2.4.2])**.** *Let X be a computable random variable in a computable metric space S. Then its distribution $\mathbf{P}_{\mathsf{X}} = \mathbf{P} \circ \mathsf{X}^{-1}$ is a computable point in the computable metric space $\mathcal{M}_1(S)$.*

On the other hand, one can show that, given a computable point μ in $\mathcal{M}_1(S)$, one can construct an i.i.d.-μ sequence of computable random variables in S.

Henceforth, we say that a measure $\mu \in \mathcal{M}_1(S)$ is computable when it is a computable point in $\mathcal{M}_1(S)$, considered as a computable metric space in this way. Note that the measure $\mathbf{P}$ on $\{0,1\}^{\mathbb{N}}$ is a computable probability measure.

Definition II.18 (Computable probability space [GHR10, Def. 2.4.1])**.** A **computable probability space** is a pair (S, μ) where S is a computable metric space and μ is a computable probability measure on S.

4. Almost decidable sets

Let (S, μ) be a computable probability space. The measure $\mu(A)$ of a c.e. open set A is always a c.e. real, but is not in general a computable real. It will be useful to understand those sets A whose measure $\mu(A)$ is a computable real.

Define A^{c} to be $S \setminus A$, i.e., the complement of A in S. Let B be an ideal ball and note that $\mu(B^{\mathsf{c}}) = 1 - \mu(B)$ is a co-c.e. real.

Definition II.19 (Almost decidable set [GHR10, Def. 3.1.3])**.** Let S be a computable metric space and let $\mu \in \mathcal{M}_1(S)$ be a probability distribution on S. A (Borel) measurable subset $A \subseteq S$ is said to be **μ-almost decidable** when there are two c.e. open sets U and V such that $U \subseteq A$ and $V \subseteq A^{\mathsf{c}}$ and $\mu(U) + \mu(V) = 1$.

Remark II.20. Let X be a computable random variable in a computable metric space S, and let $A \subseteq S$ be a $\mathbf{P}_{\mathsf{X}}$-almost decidable set. The event $\{\mathsf{X} \in A\}$ is a $\mathbf{P}$-almost decidable subset of $\{0,1\}^{\mathbb{N}}$.

When μ is a computable measure and A is an arbitrary c.e. open set, then $\mu(A)$ is merely a c.e. real. However, when A is a μ-almost decidable set, then $\mu(A)$ is also a co-c.e. real, and hence a computable real.

Lemma II.21 ([GHR10, Prop. 3.1.1])**.** *Let (S, μ) be a computable probability space, and let A be μ-almost decidable. Then $\mu(A)$ is a computable real.*

We now show that every c.e. open set is the union of a computable sequence of almost decidable subsets.

Lemma II.22 (Almost decidable subsets)**.** *Let (S, μ) be a computable probability space and let V be a c.e. open set. Then, uniformly in V, we*

can compute a sequence of μ-almost decidable sets $\{V_k\}_{k\in\mathbb{N}}$ such that, for each k, we have $V_k \subseteq V_{k+1}$ and $\bigcup_{k\in\mathbb{N}} V_k = V$.

PROOF. Note that the finite union or intersection of almost decidable sets is almost decidable. By [GHR10, Thm. 3.1.2] there is a computable sequence $\{r_j\}_{j\in\mathbb{N}}$ of reals, dense in $\mathbb{R}^+$ and for which the balls $\{B(d_i, r_j)\}_{i,j\in\mathbb{N}}$ form a basis of μ-almost decidable sets. Let $E \subseteq \mathbb{N}$ be a c.e. set such that $V = \bigcup_{i\in E} B_i$, where $\{B_i\}_{i\in\mathbb{N}}$ is the enumeration of the ideal balls of S. Consider the set $F = \{(i,j) : \exists k \in E \text{ with } \overline{B}(d_i, r_j) \subseteq B_k\}$ of indices (i,j) such that the closure of the ball $B(d_i, r_j)$ lies strictly within an ideal ball within V. Then F is c.e. and, by the density of the sequence $\{r_j\}$, we have $V = \bigcup_{(i,j)\in F} B(d_i, r_j)$. Consider the finite union $V_k := \bigcup_{\{(i,j)\in F\,:\, i,j\leq k\}} B(d_i, r_j)$, which is almost decidable. By construction, for each k, we have $V_k \subseteq V_{k+1}$ and $\bigcup_{k\in\mathbb{N}} V_k = V$. □

(A notion related to that of an almost decidable set is used for a purpose similar to Lemma II.22 in [Bos08, Cor. 2.16].)

The following converse to Lemma II.21 follows from Lemma II.22:

Corollary II.23. *Let S be a computable metric space and let $\mu \in \mathcal{M}_1(S)$ be a probability measure on S. Then μ is computable if the measure $\mu(A)$ of every μ-almost decidable set A is a computable real, uniformly in A.*

PROOF. Let V be a c.e. open set of S. By Theorem II.16, it suffices to show that $\mu(V)$ is a c.e. real, uniformly in V. By Lemma II.22, we can compute a nested sequence $\{V_k\}_{k\in\mathbb{N}}$ of μ-almost decidable sets whose union is V. Because V is open, $\mu(V) = \sup_{k\in\mathbb{N}} \mu(V_k)$. By hypothesis, $\mu(V_k)$ is a computable real for each k, and so the supremum is a c.e. real, as desired. □

Remark II.24. Let V and $\{V_k\}_{k\in\mathbb{N}}$ be as in Lemma II.22. Because V is open, $\mu(V) = \sup_{k\in\mathbb{N}} \mu(V_k)$. Hence, when $\mu(V) > 0$, we can compute a finite index k_0 such that $\mu(V_k) > 0$ for all $k \geq k_0$.

We now proceed to a study of conditional distributions.

CHAPTER III

Conditional distributions

> As yet there are no algorithms to calculate conditional expectations.
>
> M. M. Rao.
> *Conditional Measures and Applications.*
> [Rao93, Rao05]

In this chapter, we study the problem of computing conditional probabilities, a fundamental operation in probability theory and its application to statistics and machine learning. In the elementary discrete setting, a ratio of probabilities defines conditional probability. In the abstract setting, conditional probability is defined axiomatically and the search for more constructive definitions is the subject of a rich literature in probability theory and statistics. In the discrete or dominated setting, under suitable computability hypotheses, conditional probabilities are computable. However, we show that in general one cannot compute conditional probabilities. We do this by constructing a pair of computable random variables in the unit interval whose conditional distribution encodes the halting problem at every point. We show that this result is tight, in the sense that given an oracle for the halting problem, one *can* compute this conditional distribution. On the other hand, we show that conditioning in abstract settings is computable in the presence of certain additional structure, such as independent absolutely continuous noise with a computable distribution.

1. Computational limits of probabilistic inference

The use of probability to reason about uncertainty is fundamental to modern science and AI, and the computation of conditional probabilities, in order to perform evidential reasoning in probabilistic models, is perhaps its single most important computational problem.

In probabilistic programming languages, conditioning is implemented by an algorithm that accepts as input (1) a probabilistic program that samples from a joint distribution over some collection of random variables; and (2) observations for some subset of those random variables. The aim of the algorithm is to produce samples (exact, or approximate in some cases) from the conditional distribution of the unobserved random variables given the observed values. (Note that, for a particular probabilistic program and set of observations, the resulting conditioning algorithm is itself a probabilistic program, which represents the conditional distribution.) In this chapter, we focus on the simplest possible case, where a probabilistic program describes a pair of random variables, one of which is observed. The goal is to compute the associated conditional distribution.

For an experiment with a discrete set of outcomes, computing conditional probabilities is straightforward. However, in the probabilistic programming setting, it is common to place distributions on continuous or higher-order objects, and so one is already in a situation where elementary notions of conditional probability are insufficient and more sophisticated measure-theoretic notions are required. When conditioning on a continuous random variable, each particular observation has probability 0, and the elementary rule that characterizes the discrete case does not apply. Kolmogorov [Kol33] gave an axiomatic (but non-constructive) characterization of conditional probabilities. In some situations, e.g., when joint densities exist, conditioning can proceed using a continuous version of the classic Bayes' rule; however, it is not uncommon in the probabilistic programming setting for these rules to be inapplicable, and in general it is not decidable whether they are applicable. The probability and statistics literature contains many ad-hoc rules for calculating conditional probabilities in special circumstances, but even the most constructive definitions (e.g., those due to Tjur [Tju74], [Tju75], [Tju80], Pfanzagl [Pfa79], and Rao [Rao88], [Rao05]) are often not sensitive to issues of computability.

1.1. Summary of results. In Proposition III.28, we construct a pair (X, C) of computable random variables such that every version of the conditional distribution $\mathbf{P}[\mathsf{C}|\mathsf{X}]$ is discontinuous even when restricted to a $\mathbf{P}_{\mathsf{X}}$-measure one subset. (We make these notions precise in Section 3.) Every function computable on a domain D is continuous on D, and so this construction rules out the possibility of a completely general algorithm for conditioning. A natural question is whether conditioning is a computable operation when we restrict to random variables for which

some version of the conditional distribution is continuous everywhere, or at least on a measure one set.

Our main result, Theorem III.34, states that conditioning is not a computable operation on computable random variables, even in this restricted setting. We construct a pair (X, N) of computable random variables such that there is a version of the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ that is continuous on a measure one set, but no version of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ is computable. Moreover, if some oracle A computes $\mathbf{P}[\mathsf{N}|\mathsf{X}]$, then A computes the halting problem. In Theorem III.40 we strengthen this result by constructing a pair of computable random variables whose conditional distribution is noncomputable but has an *everywhere continuous* version.

This has direct implications for probabilistic programming languages. Existing proposals for such languages give algorithms for conditioning only in special cases, such as on observations of computable discrete random variables. As a corollary of our main result, there is no possible algorithm that can extend this functionality to computable *continuous* random variables.

We also characterize several circumstances in which conditioning *is* a computable operation. Under suitable computability hypotheses, conditioning is computable in the discrete setting (Lemma III.16) and where there is a conditional density (Corollary III.24).

Finally, we characterize the following situation in which conditioning on noisy data is possible. Let U, V and E be computable random variables, and define $\mathsf{Y} = \mathsf{U} + \mathsf{E}$. Suppose that $\mathbf{P}_{\mathsf{E}}$ is absolutely continuous with a computable density p_{E} (and bound M) and E is independent of U and V. In Corollary III.25, we show that the conditional distribution $\mathbf{P}[\mathsf{U}, \mathsf{V} \mid \mathsf{Y}]$ is computable.

1.2. Related work. Conditional probabilities for distributions on finite sets of discrete strings are manifestly computable, but may not be efficiently so. In this finite discrete setting, there are already interesting questions of computational complexity, which have been explored through extensions of Levin's theory of average-case complexity [Lev86]. If f is a one-way function, then it is difficult to sample from the conditional distribution of the uniform distribution of strings of some length with respect to a given output of f. This intuition is made precise by Ben-David, Chor, Goldreich, and Luby [BCGL92] in their theory of polynomial-time samplable distributions, which has since been extended by Yamakami [Yam99] and others. Extending these complexity results to the richer setting considered here is an important problem

for theoretical computer science, with repercussions for the practice of statistical AI and machine learning.

Osherson, Stob, and Weinstein [OSW88] study learning theory in the setting of *identifiability in the limit* (see [Gol67] and [Put85] for more details on this setting) and prove that a certain type of "computable Bayesian" learner fails to identify the index of a computably enumerable set that is otherwise computably identifiable in the limit. From the perspective of computable analysis, this work can be interpreted as studying when certain conditional distributions are oracle computable using the halting set as an oracle, rather than simply computable. We present a close analysis of their setup that reveals that the conditional distribution of the set index given the infinite sequence is an everywhere discontinuous function, hence fundamentally not computable in the same way as our much simpler example involving a measure concentrated on the rationals or irrationals. As we argue, the more appropriate operator to study is that restricted to those random variables whose conditional distributions admit continuous (or at the least, almost continuous) versions.

Our work is distinct from the study of conditional distributions with respect to priors that are universal for partial computable functions (as defined using Kolmogorov complexity) by Solomonoff [Sol64], Zvonkin and Levin [ZL70], and Hutter [Hut07]. The computability of conditional distributions also has a rather different character in Takahashi's work on the algorithmic randomness of points defined using universal Martin-Löf tests [Tak08]. The objects with respect to which one is conditioning in these settings are typically *computably enumerable*, but not computable. In the present work, we are interested in the problem of computing conditional distributions of random variables that are *computable* (even though the conditional distribution may itself be noncomputable).

2. Computable conditional distributions

We refer the reader to Chapter II for basic notions of computability on metric spaces.

The conditional probability of an event B given another event A captures the likelihood of the event B occurring given the knowledge that the event A has already occurred. In modern probability theory, conditional probability is generally defined with respect to a random variable (or even more abstractly, a σ-algebra), rather than a single event. In particular, Kolmogorov's [Kol33] axiomatic characterization of conditional expectation can be used to define conditional probability. For a pair of random variables Y and X in a Borel space and an arbitrary

measurable space, respectively, conditional probability can, in turn, be used to define the conditional distribution of Y given X, whether the random variable X is discrete, continuous, or otherwise.

The elementary notion of a conditional probability of one event given another is defined only when the event A has non-zero probability.

Definition III.1 (Conditional probability given an event)**.** Let S be a measurable space and let $\mu \in \mathcal{M}_1(S)$ be a probability measure on S. Let $A, B \subseteq S$ be measurable sets, and suppose that $\mu(A) > 0$. Then the **conditional probability of B given A**, written $\mu(B|A)$, is defined by

$$\mu(B|A) = \frac{\mu(B \cap A)}{\mu(A)}. \tag{8}$$

Note that for any fixed measurable $A \subseteq S$ with $\mu(A) > 0$, the function $\mu(\,\cdot\,|A)$ is a probability measure. However, this notion of conditioning is well-defined only when $\mu(A) > 0$, and so is insufficient for defining the conditional probability given the event that a *continuous* random variable takes a particular value, as such an event has measure zero.

Suppose X is a random variable mapping a probability space S to a measurable space T. For a measurable subset $A \subseteq T$, we let $\{\mathsf{X} \in A\}$ denote the inverse image $\mathsf{X}^{-1}[A] = \{s \in S : \mathsf{X}(s) \in A\}$, and for $x \in T$ we similarly define the event $\{\mathsf{X} = x\}$. We will sometimes elide some brackets and parentheses; e.g., we will use the expression $\mathbf{P}\{\mathsf{Y} \in A \mid \mathsf{X} = x\}$ to denote the conditional probability $\mathbf{P}(\{\mathsf{Y} \in A\} \mid \{\mathsf{X} = x\})$.

Before we define the abstract notion of a conditional distribution, we must define the notion of a probability kernel. For more details, see, e.g., [Kal02, Ch. 3].

Suppose T is a metric space. We let $\mathcal{B}_T$ denote the Borel σ-algebra of T, i.e., the σ-algebra generated by the open balls of T (under countable unions and complements). In this chapter, measurable functions will always be with respect to the Borel σ-algebra of a metric space.

Definition III.2 (Probability kernel)**.** Let S and T be metric spaces, and let $\mathcal{B}_T$ be the Borel σ-algebra on T. A function $\kappa : S \times \mathcal{B}_T \to [0, 1]$ is called a **probability kernel (from S to T)** when

(1) for every $s \in S$, $\kappa(s, \cdot)$ is a probability measure on T; and
(2) for every $B \in \mathcal{B}_T$, $\kappa(\cdot, B)$ is a measurable function.

We now give a characterization of conditional distributions. For more details, see, e.g., [Kal02, Ch. 6].

Definition III.3 (Conditional distribution)**.** Let X and Y be random variables in metric spaces S and T, respectively, and let $\mathbf{P}_{\mathsf{X}}$ be the distribution of X. A probability kernel κ is called a **(regular) version of the conditional distribution** $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ when it satisfies

$$\mathbf{P}\{\mathsf{X} \in A, \mathsf{Y} \in B\} = \int_A \kappa(x, B)\, \mathbf{P}_{\mathsf{X}}(dx), \tag{9}$$

for all measurable sets $A \subseteq S$ and $B \subseteq T$.

Given any two measurable functions κ_1, κ_2 satisfying (9), the functions $x \mapsto \kappa_i(x, \cdot)$ need only agree $\mathbf{P}_{\mathsf{X}}$-almost everywhere. There are many *versions*, in this sense, of a conditional distribution. However, the functions $x \mapsto \kappa_i(x, \cdot)$ will agree at points of continuity in the *support* of $\mathbf{P}_{\mathsf{X}}$.

Definition III.4. Let μ be a measure on a topological space S with open sets $\mathcal{S}$. Then the **support of** μ, written $\operatorname{supp}(\mu)$, is defined to be the set of points $x \in S$ such that all open neighborhoods of x have positive measure, i.e.,

$$\operatorname{supp}(\mu) := \{x \in S : \forall B \in \mathcal{S}\ (x \in B \implies \mu(B) > 0)\}. \tag{10}$$

Lemma III.5. *Let* X *and* Y *be random variables in topological spaces* S *and* T*, respectively, let* $\mathbf{P}_{\mathsf{X}}$ *be the distribution of* X*, and suppose that* κ_1, κ_2 *are versions of the conditional distribution* $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$*. Let* $x \in S$ *be a point of continuity of both of the maps* $x \mapsto \kappa_i(x, \cdot)$ *for* $i = 1, 2$*. If* $x \in \operatorname{supp}(\mathbf{P}_{\mathsf{X}})$*, then* $\kappa_1(x, \cdot) = \kappa_2(x, \cdot)$*.*

Proof. Fix a measurable set $A \subseteq Y$ and define $g(\cdot) := \kappa_1(\cdot, A) - \kappa_2(\cdot, A)$. We know that $g = 0$ $\mathbf{P}_{\mathsf{X}}$-a.e., and also that g is continuous at x. Assume, for the purpose of contradiction, that $g(x) = \epsilon > 0$. By continuity, there is an open neighborhood B of x, such that $g(B) \in (\frac{\epsilon}{2}, \frac{3\epsilon}{2})$. But $x \in \operatorname{supp}(\mathbf{P}_{\mathsf{X}})$, hence $\mathbf{P}_{\mathsf{X}}(B) > 0$, contradicting $g = 0$ $\mathbf{P}_{\mathsf{X}}$-a.e. □

When conditioning on a discrete random variable, a version of the conditional distribution can be built using conditional probabilities.

Example III.6. Let X and Y be random variables mapping a probability space S to a measurable space T. Suppose that X is a discrete random variable with support $R \subseteq S$, and let ν be an arbitrary probability measure on T. Consider the function $\kappa : S \times \mathcal{B}_T \to [0, 1]$ given by

$$\kappa(x, B) := \mathbf{P}\{\mathsf{Y} \in B \mid \mathsf{X} = x\} \tag{11}$$

for all $x \in R$ and $\kappa(x, \cdot) = \nu(\cdot)$ for $x \notin R$. The function κ is well-defined because $\mathbf{P}\{\mathsf{X} = x\} > 0$ for all $x \in R$, and so the right hand side of Equation (11) is well-defined. Furthermore, $\mathbf{P}\{\mathsf{X} \in R\} = 1$ and so κ is characterized by Equation (11) for almost all x. Finally,

$$\int_A \kappa(x, B)\, \mathbf{P}_{\mathsf{X}}(dx) = \sum_{x \in R \cap A} \mathbf{P}\{\mathsf{Y} \in B \mid \mathsf{X} = x\}\, \mathbf{P}\{\mathsf{X} = x\} \quad (12)$$

$$= \sum_{x \in R \cap A} \mathbf{P}\{\mathsf{Y} \in B,\ \mathsf{X} = x\}, \quad (13)$$

which is equal to $\mathbf{P}\{\mathsf{Y} \in B,\ \mathsf{X} \in A\}$, and so κ is a version of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$.

3. Computable conditional distributions

Having defined the abstract notion of a conditional distribution in Section 2, we now define our notion of computability for conditional distributions. We begin by defining a notion of computability for probability kernels.

Definition III.7 (Computable probability kernel)**.** Let S and T be computable metric spaces and let $\kappa : S \times \mathcal{B}_T \to [0, 1]$ be a probability kernel from S to T. Then we say that κ is a **computable (probability) kernel** when the map $\phi_\kappa : S \to \mathcal{M}_1(T)$ given by $\phi_\kappa(s) := \kappa(s, \cdot)$ is a computable function. Similarly, we say that κ is computable on a subset $D \subseteq S$ when ϕ_κ is computable on D.

Recall that for a fixed Borel set $B \subseteq T$ and kernel $\kappa : S \times \mathcal{B}_T \to [0, 1]$, the function $\kappa(\cdot, B)$ is measurable. We now prove a corresponding property of computable probability kernels.

Lemma III.8. *Let S and T be computable metric spaces, let κ be a probability kernel from S to T computable on a subset $D \subseteq S$, and let $A \subseteq T$ be a c.e. open set. Then $\kappa(\cdot, A) : S \to [0, 1]$ is lower semicomputable on D, i.e., for every rational $q \in (0, 1)$ there is a c.e. open set V_q uniformly computable in q and A, such that $\kappa(\cdot, A)^{-1}[(q, 1]] \cap D = V_q \cap D$.*

Proof. Let ϕ_κ be as in Definition III.7, fix a rational $q \in (0, 1)$ and c.e. open set A, and define $I = (q, 1]$. Then $\kappa(\cdot, A)^{-1}[I] = \phi_\kappa^{-1}[P]$, where

$$P = \{\mu \in \mathcal{M}_1(T) \,:\, \mu(A) > q\}. \quad (14)$$

This is an open set in the weak topology induced by the Prokhorov metric (see [Sch07, Lem. 3.2]). We now show that P is, in fact, c.e. open.

Consider the set $\mathcal{D}$ of all probability measures on $(T, \delta_T, \mathcal{D}_T)$ that are concentrated on a finite subset and where the measure of each atom is rational, i.e., every $\nu \in \mathcal{D}$ can be written as $\nu = \sum_{i=1}^{k} q_i \delta_{t_i}$ for some rationals $q_i \geq 0$ such that $\sum_{i=1}^{k} q_i = 1$ and some points $t_i \in \mathcal{D}_T$. Gács [Gác05, §B.6.2] shows that $\mathcal{D}$ is dense in the Prokhorov metric and makes $\mathcal{M}_1(T)$ a computable metric space.

Let $\nu \in \mathcal{D}$ be concentrated on the finite set R. Gács [Gác05, Prop. B.17] characterizes the ideal ball E centered at ν with rational radius $\epsilon > 0$ as the set of measures $\mu \in \mathcal{M}_1(T)$ such that

$$\mu(C^\epsilon) > \nu(C) - \epsilon \tag{15}$$

for all $C \subseteq R$, where $C^\epsilon = \bigcup_{t \in C} B(t, \epsilon)$.

We can write $A = \bigcup_{n \in \mathbb{N}} B(d_n, r_n)$ for a computable sequence of ideal balls in T with centers d_n and rational radii r_n. It follows that $E \subseteq P$ if and only if $\nu(R \cap A^{-\epsilon}) > q$, where $A^{-\epsilon} := \bigcup_{n \in \mathbb{N}} B(d_n, r_n - \epsilon)$. Note $A^{-\epsilon}$ is again c.e. open. As ν is concentrated on R, it follows that $\nu(R \cap A^{-\epsilon})$ is a c.e. real and so we can semidecide whether $E \subseteq P$. Therefore, P is a c.e. open set.

Hence, by the computability of ϕ_κ, there is a c.e. open set V, uniformly computable in P (and hence I) such that $\phi_\kappa^{-1}[P] \cap D = V \cap D$. But then, we have that $\kappa(\cdot, A)^{-1}[I] \cap D = V \cap D$, and so $\kappa(\cdot, A)$ is computable on D. □

In fact, the lower semicomputability of $\kappa(\cdot, A) : S \to [0, 1]$ on $D \subseteq S$ for arbitrary c.e. open sets $A \subseteq T$ is equivalent to being able to compute $\phi_\kappa : S \to \mathcal{M}_1(T)$ on D.

Lemma III.9. *Let S and T be computable metric spaces, let κ be a probability kernel from S to T, and let $D \subseteq S$. Then $\kappa(\cdot, A)$ is lower semicomputable on $D \subseteq S$ uniformly in a c.e. open set A if and only if ϕ_κ is computable on D.*

Proof. The implication from right to left was shown in Lemma III.8. Now assume that $\kappa(\cdot, A)$ is lower semicomputable on D uniformly in A. In other words, uniformly in $s \in D$ and in A, we have that $\kappa(s, A)$ is a c.e. real. But by Lemma II.16, this implies that $\kappa(s, \cdot)$ is computable on D. □

Hence one may naturally interpret a computable probability kernel κ as either a computable probability measure $\kappa(s, \cdot)$ for each computable $s \in S$, or as a lower semicomputable function $\kappa(\cdot, A)$ for each c.e. open set $A \subseteq T$. In fact, when $A \subseteq T$ is a decidable set (i.e., A and $T \setminus A$ are both c.e. open), $\kappa(\cdot, A)$ is a computable function.

Corollary III.10. *Let S and T be computable metric spaces, let κ be a probability kernel from S to T computable on a subset $D \subseteq S$, and let $A \subseteq T$ be a decidable set. Then $\kappa(\cdot, A) : S \to [0,1]$ is computable on D.*

PROOF. If B a c.e. open set, $\kappa(\cdot, B)$ is lower semicomputable on D and $\kappa(\cdot, T \setminus B) = 1 - \kappa(\cdot, B)$ is upper semicomputable on D. Because A is decidable, both A and $T \setminus A$ are c.e. open, and so $\kappa(\cdot, A)$ is computable on D. □

Having pinned down a notion of computability for probability kernels, we now return to the question of computability for conditional distributions. The first apparent obstacle is the fact that a conditional distribution may have many different versions. However, their computability as probability kernels does not differ (up to a change in domain by a null set).

Lemma III.11. *Let κ be a version of a conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ that is computable on some $\mathbf{P}_\mathsf{X}$-measure one set. Then any version of $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is also computable on some $\mathbf{P}_\mathsf{X}$-measure one set.*

PROOF. Let κ be a version that is computable on a $\mathbf{P}_\mathsf{X}$-measure one set D, and let κ' be any other version. Then $Z := \{s \in S : \kappa(s, \cdot) \neq \kappa'(s, \cdot)\}$ is a $\mathbf{P}_\mathsf{X}$-null set, and $\kappa = \kappa'$ on $D \setminus Z$. Hence κ' is computable on the $\mathbf{P}_\mathsf{X}$-measure one set $D \setminus Z$. □

These results suggest the following definition of computability for conditional distributions.

Definition III.12 (Computable conditional distributions)**.** We say that the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is computable when there is some version κ that is computable on a $\mathbf{P}_\mathsf{X}$-measure one subset of S.

Intuitively, a conditional distribution is computable when for some (and hence for any) version κ there is a program that, given as input a representation of a point $s \in S$, outputs a representation of the measure $\phi_\kappa(s) = \kappa(s, \cdot)$ for $\mathbf{P}_\mathsf{X}$-almost all inputs s.

Suppose that $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is computable, i.e., there is a version κ for which the map ϕ_κ is computable on some $\mathbf{P}_\mathsf{X}$-measure one set $S' \subseteq S$. (As noted in Definition III.7, we will often abuse notation and say that κ is computable on S'.) Because a function is computable on a subset only if it is continuous on that subset, the restriction of ϕ_κ to S' is necessarily continuous (under the subspace topology on S'). We will say that κ is **$\mathbf{P}_\mathsf{X}$-almost continuous** when the restriction of ϕ_κ to some $\mathbf{P}_\mathsf{X}$-measure one set is continuous. (Note that this does not necessarily imply $\mathbf{P}_\mathsf{X}$-a.e. continuity.) Thus when $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is computable, there is some $\mathbf{P}_\mathsf{X}$-almost continuous version.

In Section 4 we describe a pair of computable random variables X, Y for which $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is not computable, by virtue of no version being $\mathbf{P}_{\mathsf{X}}$-almost continuous. In Section 5 we describe a pair (X, N) of computable random variables for which there is a $\mathbf{P}_{\mathsf{X}}$-almost continuous version of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$, yet no version that is computable on a $\mathbf{P}_{\mathsf{X}}$-measure one set.

3.1. Discrete setting. First we study situations where conditioning on a discrete random variable is computable.

Recall the definition of conditional probability (Definition III.1). When μ is computable and A is an almost decidable set, the conditional probability given A is computable.

Lemma III.13 (Conditional probability given an almost decidable set [GHR10, Prop. 3.1.2])**.** *Let (S, μ) be a computable probability space and let A be an almost decidable subset of S satisfying $\mu(A) > 0$. Then $\mu(\cdot|A)$ is computable.*

Proof. By Corollary II.23, it suffices to show that $\frac{\mu(B \cap A)}{\mu(A)}$ is computable for an almost decidable set B. But then $B \cap A$ is almost decidable and so its measure, the numerator, is a computable real. The denominator is likewise the measure of an almost decidable set, hence a computable real. Finally, the ratio of two computable reals is computable. □

The equation

$$\mathbf{P}\{\mathsf{Y} \in A \mid \mathsf{X} = x\} = \frac{\mathbf{P}\{\mathsf{Y} \in A,\ \mathsf{X} = x\}}{\mathbf{P}\{\mathsf{X} = x\}} \tag{16}$$

gives a recipe for calculating the conditional distribution of a discrete random variable. However, the event $\{\mathsf{X} = x\}$ is not necessarily even an open set, and so in order to compute the conditional distribution given a discrete random variable, we need additional computability hypotheses on its support.

Definition III.14 (Computably discrete set)**.** Let S be a computable metric space. We say that a (finite or countably infinite) subset $D \subseteq S$ is **computably discrete** when, for some enumeration $d_0, d_1, \ldots$ of D (possibly with repetition) there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that each d_j is the unique point of D in the ideal ball $B_{f(j)}$.

The following result follows immediately from Lemma II.22.

Lemma III.15. *Let (S, μ) be a computable probability space, and let D be a computably discrete subset of S. Define $D_+ := \{d \in D : \mu(\{d\}) > 0\}$. There is a partial function $g : S \to \mathbb{N}$, computable on D_+, such*

that for $d \in D_+$, the integer $g(d)$ is the index of a μ-almost decidable set containing d and no other points of D.

Lemma III.16 (Conditioning on a discrete random variable)**.** *Let X and Y be computable random variables in computable metric spaces S and T, respectively. Assume that $\mathbf{P}_{\mathsf{X}}$ is concentrated on a computably discrete set D. Then the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is computable, uniformly in X, Y and D.*

Proof. Define $D_+ := \{d \in D : \mathbf{P}_{\mathsf{X}}(d) > 0\}$, and let g be a computable partial function that assigns each point in D_+ a $\mathbf{P}_{\mathsf{X}}$-almost decidable set covering it, as in Lemma III.15. Let $A_{g(d)}$ denote the $\mathbf{P}_{\mathsf{X}}$-almost decidable set coded by $g(d)$.

Because X is also concentrated on D_+, a version κ of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is an arbitrary kernel $\kappa(\cdot,\cdot)$ that satisfies

$$\kappa(d,\cdot) = \mathbf{P}\{\mathsf{Y} \in \cdot \mid \mathsf{X} = d\} \tag{17}$$

for every $d \in D_+$ (as in Example III.6).

Let $d \in D_+$ be arbitrary. The set $A_{g(d)}$ contains exactly one point of positive $\mathbf{P}_{\mathsf{X}}$-measure, and so the events $\{\mathsf{X} = d\}$ and $\{\mathsf{X} \in A_{g(d)}\}$ are positive $\mathbf{P}_{\mathsf{X}}$-measure sets that differ by a $\mathbf{P}_{\mathsf{X}}$-null set. Hence

$$\mathbf{P}\{\mathsf{Y} \in \cdot \mid \mathsf{X} = d\} = \mathbf{P}\{\mathsf{Y} \in \cdot \mid \mathsf{X} \in A_{g(d)}\}. \tag{18}$$

By Remark II.20, the event $\{\mathsf{X} \in A_{g(d)}\}$ is $\mathbf{P}$-almost decidable, and so the measure $\mathbf{P}\{\mathsf{Y} \in \cdot \mid \mathsf{X} \in A_{g(d)}\}$ is computable, by Lemma III.13.

Thus the partial function mapping $S \to \mathcal{M}_1(T)$ by

$$x \mapsto \mathbf{P}\{\mathsf{Y} \in \cdot \mid \mathsf{X} \in A_{g(x)}\} \tag{19}$$

is computable on D_+, a subset of S of $\mathbf{P}_{\mathsf{X}}$-measure one, and so the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is computable. □

3.2. Continuous, dominated, and other settings. Although we show that the general case of conditioning on a random variable is not computable, additional structure can sometimes make conditioning computable. For example, in Chapter IV, Section 6 we study posterior inference in the setting of exchangeable sequences, and give a positive result covering a wide class of Bayesian nonparametric models.

Another common situation where we know the form of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$ is when the conditional distribution $\mathbf{P}[\mathsf{X}|\mathsf{Y}]$ is *dominated*, i.e., when there exists a conditional density.

We recall the following standard definitions of density and conditional density.

Definition III.17 (Density)**.** Let $(\Omega, \mathcal{A}, \nu)$ be a measure space and let $f : A \to \mathbb{R}^+$ be a measurable function. Then the function μ on $\mathcal{A}$ given by

$$\mu(A) = \int_A f d\nu \tag{20}$$

for $A \in \mathcal{A}$ is a measure on $(\Omega, \mathcal{A})$ and f is called a **density of μ with respect to ν**. Note that g is a density of μ with respect to ν if and only if $f = g$ ν-a.e.

Definition III.18 (Conditional density)**.** Let X and Y be random variables in metric spaces S and T, respectively, let $\kappa_{\mathsf{X}|\mathsf{Y}}$ be a version of the conditional distribution $\mathbf{P}[\mathsf{X}|\mathsf{Y}]$, and assume that there exists a measure $\nu \in \mathcal{M}(S)$ and measurable function $p_{\mathsf{X}|\mathsf{Y}}(x|y) : S \times T \to \mathbb{R}^+$ such that $p_{\mathsf{X}|\mathsf{Y}}(\cdot|y)$ is a density of $\kappa_{\mathsf{X}|\mathsf{Y}}(y, \cdot)$ with respect to ν for $\mathbf{P}_{\mathsf{Y}}$-a.e. y. That is,

$$\kappa_{\mathsf{X}|\mathsf{Y}}(y, A) = \int_A p_{\mathsf{X}|\mathsf{Y}}(x|y)\nu(dx) \tag{21}$$

for measurable sets $A \subseteq S$ and $\mathbf{P}_{\mathsf{Y}}$-almost all y. Then $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ is called a **conditional density (with respect to ν) of X given Y**.

Common parametric families of distributions (e.g., exponential families like Gaussian, Gamma, etc.) admit conditional densities, and in these cases, the well-known Bayes' rule (22) gives a formula for expressing the conditional distribution.

Lemma III.19 (Bayes' rule [Sch95, Thm. 1.13])**.** *Let X and Y be random variables as in Definition III.3, let $\kappa_{\mathsf{X}|\mathsf{Y}}$ be a version of the conditional distribution $\mathbf{P}[\mathsf{X}|\mathsf{Y}]$, and assume that there exists a* conditional density *$p_{\mathsf{X}|\mathsf{Y}}(x|y)$ with respect to $\nu \in \mathcal{M}_1(S)$. Then the function given by*

$$\kappa_{\mathsf{Y}|\mathsf{X}}(x, B) := \frac{\int_B p_{\mathsf{X}|\mathsf{Y}}(x|y)\mathbf{P}_{\mathsf{Y}}(dy)}{\int p_{\mathsf{X}|\mathsf{Y}}(x|y)\mathbf{P}_{\mathsf{Y}}(dy)}, \tag{22}$$

is a version of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$. □

Remark III.20. Comparing Equation (21) to (22), we see that

$$p_{\mathsf{Y}|\mathsf{X}}(y|x) = \frac{p_{\mathsf{X}|\mathsf{Y}}(x|y)}{\int p_{\mathsf{X}|\mathsf{Y}}(x|y)\mathbf{P}_{\mathsf{Y}}(dy)} \tag{23}$$

is the conditional density with respect to $\mathbf{P}_{\mathsf{Y}}$ of Y given X.

Lemma III.21. *Let $R \subseteq S$ be a $\mathbf{P}_{\mathsf{X}}$-measure one subset. If the conditional density $p_{\mathsf{X}|\mathsf{Y}}(x|y)$ of X given Y is continuous on $R \times T$ and*

bounded, then there is a $\mathbf{P}_{\mathsf{X}}$*-almost continuous version of the conditional distribution* $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$.

PROOF. Fix an open set $B \subseteq T$. We will show that for fixed B, the map $x \mapsto \kappa_{\mathsf{Y}|\mathsf{X}}(x, B)$ given by Equation (22) is a lower semicontinuous by demonstrating that the numerator is lower semicontinuous, while the denominator is continuous.

Let $\mathbf{P}_{\mathsf{Y}}$ be the distribution of Y. By hypothesis, the map $\phi : S \to \mathcal{C}(T, \mathbb{R}^+)$ given by $\phi(x) = p_{\mathsf{X}|\mathsf{Y}}(x|\cdot)$ is continuous on R, while the indicator function $\mathbf{1}_B$ is lower semicontinuous. Because the integration operator $f \mapsto \int f d\mu$ of a lower semicontinuous function f with respect to a probability measure μ is itself lower semicontinuous, the map $x \mapsto \int \mathbf{1}_B \phi(x) d\mathbf{P}_{\mathsf{Y}}$ is lower semicontinuous on R.

Now let $B = T$ and note that for every $x \in R$, the function $\phi(s)$ is bounded by hypothesis. Integration of a bounded continuous function with respect to a probability measure is a continuous operation, and so the map $x \mapsto \int \phi(x) d\mathbf{P}_{\mathsf{Y}}$ is continuous on R. Therefore, $\kappa_{\mathsf{Y}|\mathsf{X}}$ is $\mathbf{P}_{\mathsf{X}}$-almost continuous. □

Example III.22. Let Y be a Gaussian random variable with zero mean and unit variance, and, conditioned on Y, let X be a Gaussian random variable with mean Y and unit variance. Then the conditional density (with respect to Lebesgue measure) is given by $p_{\mathsf{X}|\mathsf{Y}}(x|y) = (2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}(x-y)^2}$. Furthermore, the conditional density of Y given X (with respect to Lebesgue measure) is given by $p_{\mathsf{Y}|\mathsf{X}}(y|x) = (\pi)^{-\frac{1}{2}} e^{-\frac{1}{4}(x-2y)^2}$, which implies that Y, conditioned on X, is Gaussian distributed with mean $\frac{\mathsf{X}}{2}$ and variance $\frac{1}{2}$.

We now turn to the question of computing conditional distributions in the dominated setting. We begin by reviewing a well-known integration result.

Proposition III.23 (Integration of computable functions ([HR09, Cor. 4.3.2]))**.** *Let* (S, μ) *be a computable probability space. Let* $f : S \to \mathbb{R}^+$ *be a computable function and let* M *be a bound on* f*. Then* $\int f d\mu$ *is a computable real, uniformly in* f *and* M*.*

Using this result, we can now study when the conditional distribution characterized by Equation (22) is computable.

Corollary III.24 (Density and independence)**.** *Let* U*,* V*, and* Y *be computable random variables (in computable metric spaces), where* Y *is independent of* V *given* U*. Assume that there exists a conditional density* $p_{\mathsf{Y}|\mathsf{U}}(y|u)$*, and furthermore that* $p_{\mathsf{Y}|\mathsf{U}}$ *and a bound* M *for* $p_{\mathsf{Y}|\mathsf{U}}$ *are computable. Then the conditional distribution* $\mathbf{P}[(\mathsf{U}, \mathsf{V})|\mathsf{Y}]$ *is computable.*

PROOF. Let $\mathsf{X} = (\mathsf{U}, \mathsf{V})$. The conditional density $p_{\mathsf{Y}|\mathsf{X}}(y|x)$ of Y given X exists and satisfies

$$p_{\mathsf{Y}|\mathsf{X}}(y|(u,v)) = c(y)\, p_{\mathsf{Y}|\mathsf{U}}(y|u), \tag{24}$$

for some measurable function c. Note that the $c(y)$ term cancels when the ratio is taken as in Equation (22). Therefore, the computability of the integrand and the bound imply by Proposition III.23 that a kernel is computable. □

As an immediate corollary, we obtain the computability of conditioning in the following common situation in probabilistic modeling: where the observed random variable has been corrupted by independent absolutely continuous noise.

Corollary III.25 (Independent noise)**.** *Let* V *be a computable random variable in a computable metric space and let* U *and* E *be computable random variables in* $\mathbb{R}$*. Define* $\mathsf{Y} = \mathsf{U} + \mathsf{E}$*. If* $\mathbf{P}_{\mathsf{E}}$ *is absolutely continuous with a computable density* p_{E} *(and bound* M*) and* E *is independent of* U *and* V *then the conditional distribution* $\mathbf{P}[(\mathsf{U}, \mathsf{V}) \mid \mathsf{Y}]$ *is computable.*

PROOF. The conditional density $p_{\mathsf{Y}|\mathsf{U}}(y|u)$ of Y given U exists and satisfies $p_{\mathsf{Y}|\mathsf{U}}(y|u) = p_{\mathsf{E}}(y - u)$. The result then follows from Corollary III.24. □

It follows from our main noncomputability result (Theorem III.34) that noiseless observations cannot always be computably approximated by noisy ones. For example, even though an observation corrupted with zero mean Gaussian noise with standard deviation σ may recover the original condition as $\sigma \to 0$, Theorem III.34 implies that one cannot, in general, compute how small σ must be in order to bound the error introduced by noise.

By Myhill [Myh71], there is a computable function $[0,1] \to \mathbb{R}$ whose derivative is continuous, but not computable. However, Pour-El and Richards [PER89, Ch. 1, Thm. 2] show that all twice continuously differentiable computable functions from $[0,1] \to \mathbb{R}$ have computable derivatives. Therefore, noise with a sufficiently smooth probability distribution has a computable density, and by Corollary III.25, a computable random variable corrupted by such noise still admits a computable conditional distribution.

This result is analogous to a classical theorem of information theory. Hartley [Har28] and Shannon [Sha49] show that the capacity of a continuous real-valued channel without noise is infinite, yet the addition of, e.g., independent Gaussian noise with $\epsilon > 0$ variance causes the channel capacity to become finite. The Gaussian noise prevents an

infinite amount of information from being encoded in the bits of the real number. Similarly, it is not possible in general to incorporate the information in a continuous observation when computing a conditional probability. However, the addition of sufficiently smooth and independent computable noise makes conditioning possible on a (finite) computer.

3.3. Conditional distributions as limits. Without additional structure like discreteness or a conditional density, one must rely on the abstract definition of conditional probability. Unfortunately, it does not immediately suggest a method of computation. We now introduce a constructive definition of conditional distributions due to Tjur. More details can found in [Tju80, §9.7]. Pfanzagl [Pfa79] describes a somewhat similar approach.

Roughly speaking, a set B is near a point x if some neighborhood of x contains B. By shrinking such neighborhoods, this allows for a notion of convergence of a directed system of sets to a point x that allows the sets to vary freely within the shrinking neighborhoods. Given random variables X, Y, one can then take the limit (according to this notion of convergence) of the discrete conditional distributions of Y given $\mathsf{X} \in B$ to obtain a version of the conditional distribution of Y given $\mathsf{X} = x$.

Definition III.26 (Tjur Property [Tju75])**.** Let X and Y be random variables in complete metric spaces, and let $x \in \operatorname{supp}(\mathbf{P}_{\mathsf{X}})$. Let $\mathscr{D}(x)$ denote the set of pairs (V, B) where V is an open neighborhood of x and B is a measurable subset of V with $\mathbf{P}_{\mathsf{X}}(B) > 0$. We say that a pair (V, B) **is closer to** x **than** (V', B') if $V' \supseteq V$. Note that this relation is a partial ordering on $\mathscr{D}(x)$ and makes $\mathscr{D}(x)$ a directed set. We say that x **has the Tjur property (for** Y **given** X**)** when the directed limit

$$\mathbf{P}^x_{\mathsf{Y}}(\cdot) := \lim_{(V,B)\in\mathscr{D}(x)} \mathbf{P}_{\mathsf{Y}}(\ \cdot\ \mid \mathsf{X} \in B) \tag{25}$$

exists and is a probability measure.

Many common properties imply that a point is Tjur. For example, for an absolutely continuous random variable with density f, every point x of continuity of f such that $f(x) > 0$ is a Tjur point. Also, any isolated point mass (e.g., a point in the support of a discrete random variable) is a Tjur point. On the other hand, nonisolated point masses are not necessarily Tjur points.

Tjur points sometimes exist even in nondominated settings. Let G be a Dirichlet process with an absolutely continuous base measure H on a computable metric space S. That is, G is a random discrete

probability distribution on S. Conditioned on G, let X be G-distributed (i.e., X is a sample from the random distribution G). Then any point $x \in S$ in the support of H is a Tjur point, yet there does not exist a conditional density of G given X.

The following lemma is a consequence of Corollary 9.9.2 and Proposition 9.10.1 of [Tju80].

Lemma III.27. *Let X and Y be random variables in complete metric spaces S and T, and suppose that $\mathbf{P}_{\mathsf{X}}$-almost all $x \in S$ have the Tjur property (for Y given X). For each Tjur point x, suppose that $\{B_n^x\}_{n\in\mathbb{N}}$ is a sequence of measurable sets for which each B_n^x is contained in the 2^{-n}-ball around x and $\mathbf{P}(B_n^x) > 0$.*

Then the function $\kappa : S \times \mathcal{B}_T \to [0,1]$ given by

$$\kappa(x, A) := \lim_{n\to\infty} \mathbf{P}\{\mathsf{Y} \in A \mid \mathsf{X} \in B_n^x\} \tag{26}$$

for Tjur points x and Borel sets $A \subseteq T$ (and defined by $\kappa(x,\cdot) = \nu$ for an arbitrary probability measure ν otherwise) is a version of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$.

For example, if X and Y are real random variables, then taking $B_n^x := (x - 2^{-n}, x + 2^{-n})$ in (26) gives a version of the conditional distribution $\mathbf{P}[\mathsf{Y}|\mathsf{X}]$. In Section 7, we take B_n^x to be a sequence of almost decidable sets.

Even when such limits exist, they may not be computable. In fact, the main construction in Section 5 is an example of a conditional distribution for which almost all points are Tjur, and yet no version of the conditional distribution is a computable measure.

Before proceeding to the main construction, in Section 4 we first present a pair of computable random variables for which no point is Tjur.

4. Discontinuous conditional distributions

Any attempt to characterize the computability of conditional distributions immediately runs into the following roadblock: a conditional distribution need not have *any* version that is continuous or even almost continuous (in the sense described in Section 3).

Recall that a random variable C is a **Bernoulli**(p) random variable, or equivalently, a **p-coin**, when $\mathbf{P}\{\mathsf{C} = 1\} = 1 - \mathbf{P}\{\mathsf{C} = 0\} = p$. We call a $\frac{1}{2}$-coin a **fair coin**. A random variable N is **geometric** when it takes values in $\mathbb{N} = \{0, 1, 2, \dots\}$ and satisfies $\mathbf{P}\{\mathsf{N} = n\} = 2^{-(n+1)}$, for $n \in \mathbb{N}$. A random variable that takes values in a discrete set is a **uniform** random variable when it assigns equal probability to each element. A

continuous random variable U on the unit interval is **uniform** when the probability that it falls in the subinterval $[\ell, r]$ is $r - \ell$. It is easy to show that the distributions of these random variables are computable.

Let C, U, and N be independent computable random variables, where C is a fair coin, U is a uniform random variable on $[0, 1]$, and N is a geometric random variable. Fix a computable enumeration $\{r_i\}_{i\in\mathbb{N}}$ of the rational numbers (without repetition) in $(0, 1)$, and consider the random variable

$$\mathsf{X} := \begin{cases} \mathsf{U}, & \text{if } \mathsf{C} = 1; \\ r_{\mathsf{N}}, & \text{otherwise.} \end{cases} \tag{27}$$

It is easy to verify that X is a computable random variable.

Proposition III.28. *No version of the conditional distribution* $\mathbf{P}[\mathsf{C}|\mathsf{X}]$ *is* $\mathbf{P}_{\mathsf{X}}$*-almost continuous.*

Proof. Note that $\mathbf{P}\{\mathsf{X} \text{ rational}\} = \frac{1}{2}$ and, in particular, we have that $\mathbf{P}\{\mathsf{X} = r_k\} = \frac{1}{2^{k+1}} > 0$. Therefore, any two versions of the conditional distribution $\mathbf{P}[\mathsf{C}|\mathsf{X}]$ must agree on *all* rationals in $[0, 1]$. In addition, any two versions must agree on *almost all* irrationals in $[0, 1]$ because the support of U is all of $[0, 1]$. An elementary calculation shows that $\mathbf{P}\{\mathsf{C} = 0 \mid \mathsf{X} \text{ rational}\} = 1$, while $\mathbf{P}\{\mathsf{C} = 0 \mid \mathsf{X} \text{ irrational}\} = 0$. Therefore, all versions κ of $\mathbf{P}[\mathsf{C}|\mathsf{X}]$ satisfy

$$\kappa(x, \{0\}) = \begin{cases} 1, & x \text{ rational}; \\ 0, & x \text{ irrational}, \end{cases} \quad \text{a.s.}, \tag{28}$$

which, when considered as a function of x, is the *nowhere continuous* function known as the Dirichlet function.

Suppose some version κ were continuous when restricted to some $\mathbf{P}_{\mathsf{X}}$-measure one subset $D \subseteq [0, 1]$. But D must contain every rational and almost every irrational in $[0, 1]$, and so the inverse image of an open set containing 1 but not 0 would be the set of rationals, which is not open in the subspace topology induced on D. □

In some ways this construction is disappointing: a representation of the unit interval as the disjoint union of the rationals and irrationals, i.e., a representation for X which stores whether the value is rational or not, would make the conditional distribution continuous and computable. However, this representation of real numbers is very problematic: e.g., multiplication becomes discontinuous, hence not computable.

The potential discontinuity of conditional probabilities is a fundamental obstacle to the computability of conditional distributions. This suggests that we focus our attention on settings that admit almost

continuous or continuous versions. We might still hope to be able to compute the conditional distribution when there is *some* version that is almost continuous or even continuous. However we will show that even this is not possible in general.

5. Noncomputable almost continuous conditional distributions

In this section, we construct a pair of random variables (X, N) that is computable, yet whose conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ is not computable, despite the existence of a $\mathbf{P}_{\mathsf{X}}$-almost continuous version.

Let $h : \mathbb{N} \to \mathbb{N} \cup \{\infty\}$ be the map given by $h(n) = \infty$ if the nth Turing machine does not halt (on input 0) and $h(n) = k$ if the nth Turing machine halts (on input 0) at the kth step. The function h is lower-semicomputable because we can compute all lower bounds: for all $k \in \mathbb{N}$, we can run the nth TM for k steps to determine whether $h(n) < k$, or $h(n) = k$, or $h(n) > k$. But h is not computable because any finite upper bound on $h(n)$ would imply that the nth TM halts, thereby solving the halting problem. However, we will define a computable random variable X such that conditioning on its value recovers h.

Let N be a computable geometric random variable, C a computable $\frac{1}{3}$-coin and U and V both computable uniform random variables on $[0, 1]$, all mutually independent. Let $\lfloor x \rfloor$ denote the greatest integer $y \leq x$. Note that $\lfloor 2^k \mathsf{V} \rfloor$ is uniformly distributed on $\{0, 1, 2, \ldots, 2^k - 1\}$. Consider the derived random variables $\mathsf{X}_k := \frac{2\lfloor 2^k \mathsf{V} \rfloor + \mathsf{C} + \mathsf{U}}{2^{k+1}}$ for $k \in \mathbb{N}$. The limit $\mathsf{X}_\infty := \lim_{k\to\infty} \mathsf{X}_k$ exists with probability one and satisfies $\lim_{k\to\infty} \mathsf{X}_k = \mathsf{V}$ a.s. Finally, we define $\mathsf{X} := \mathsf{X}_{h(\mathsf{N})}$. The following proposition gives more insight into the meaning of these definitions.

Proposition III.29. *The random variable* X *is computable.*

Proof. Let $\{\mathsf{U}_n : n \in \mathbb{N}\}$ and $\{\mathsf{V}_n : n \in \mathbb{N}\}$ be the binary expansions of U and V, respectively. Because U and V are computable and almost surely irrational, it is not hard to show that their binary expansions are computable random variables in $\{0, 1\}$, uniformly in n.

For each $k \geq 0$, define the random variable

$$\mathsf{D}_k = \begin{cases} \mathsf{V}_k, & h(\mathsf{N}) > k; \\ \mathsf{C}, & h(\mathsf{N}) = k; \\ \mathsf{U}_{k-h(\mathsf{N})-1}, & h(\mathsf{N}) < k. \end{cases} \tag{29}$$

Because h is lower-semicomputable, $\{\mathsf{D}_k\}_{k\geq 0}$ are computable random variables, uniformly in k.[1] We now show that, with probability one, $\{\mathsf{D}_k\}_{k\geq 0}$ is the binary expansion of X, thus showing that X is itself a computable random variable.

There are two cases to consider:

First, conditioned on $h(\mathsf{N}) = \infty$, we have that $\mathsf{D}_k = \mathsf{V}_k$ for all $k \geq 0$. In fact, $\mathsf{X} = \mathsf{V}$ when $h(\mathsf{N}) = \infty$, and so the binary expansions match.

Condition on $h(\mathsf{N}) = m$ and let D denote the computable random real whose binary expansion is $\{\mathsf{D}_k\}_{k\geq 0}$. We must then show that $\mathsf{D} = \mathsf{X}_m$ a.s. Note that $\lfloor 2^m \mathsf{X}_m \rfloor = \lfloor 2^m \mathsf{V} \rfloor = \sum_{k=0}^{m-1} 2^{m-1-k} \mathsf{V}_k = \lfloor 2^m \mathsf{D} \rfloor$, and thus the binary expansions agree for the first m digits. In a similar fashion, one can show that the next binary digit of X_m is C, followed by the binary expansion U, thus agreeing with D for all $k \geq 0$. □

For a visualization of (X, N), see Figure 1.

We now show that $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ is not computable, despite the existence of a $\mathbf{P}_{\mathsf{X}}$-almost continuous version of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$. We begin by characterizing the conditional density of X given N. Note that the constant function $p_{\mathsf{X}_\infty}(x) := 1$ is the density of X_∞ with respect to Lebesgue measure on $[0, 1]$.

Lemma III.30. *For each $k \in \mathbb{N}$, the distribution of X_k admits a density p_{X_k} with respect to Lebesgue measure on $[0, 1]$ given by*

$$p_{\mathsf{X}_k}(x) = \begin{cases} \frac{4}{3}, & \lfloor 2^{k+1} x \rfloor \text{ even;} \\ \frac{2}{3}, & \lfloor 2^{k+1} x \rfloor \text{ odd.} \end{cases} \tag{30}$$

Proof. Let $k \in \mathbb{N}$. With probability one, the integer part of $2^{k+1}\mathsf{X}_k$ is $2\lfloor 2^k \mathsf{V} \rfloor + \mathsf{C}$ while the fractional part is U. Therefore, the distribution of $2^{k+1}\mathsf{X}_k$ (and hence X_k) admits a piecewise constant density with respect to Lebesgue measure.

In particular, $\lfloor 2^{k+1}\mathsf{X}_k \rfloor \equiv \mathsf{C} \pmod 2$ almost surely and $2\lfloor 2^k \mathsf{V} \rfloor$ is independent of C and uniformly distributed on $\{0, 2, \ldots, 2^{k+1} - 2\}$. Therefore,

$$\mathbf{P}\{\lfloor 2^{k+1}\mathsf{X}_k \rfloor = \ell\} = 2^{-k} \cdot \begin{cases} \frac{2}{3}, & \ell \text{ even;} \\ \frac{1}{3}, & \ell \text{ odd,} \end{cases} \tag{31}$$

[1] Informally, one can sample the bits D_k by simulating the Nth Turing machine and returning a fair coin flip on each stage for which the Turing machine does not halt or has already halted, and returning a biased coin exactly when the machine halts. In particular, if the machine never halts, then every bit is a fair coin flip.

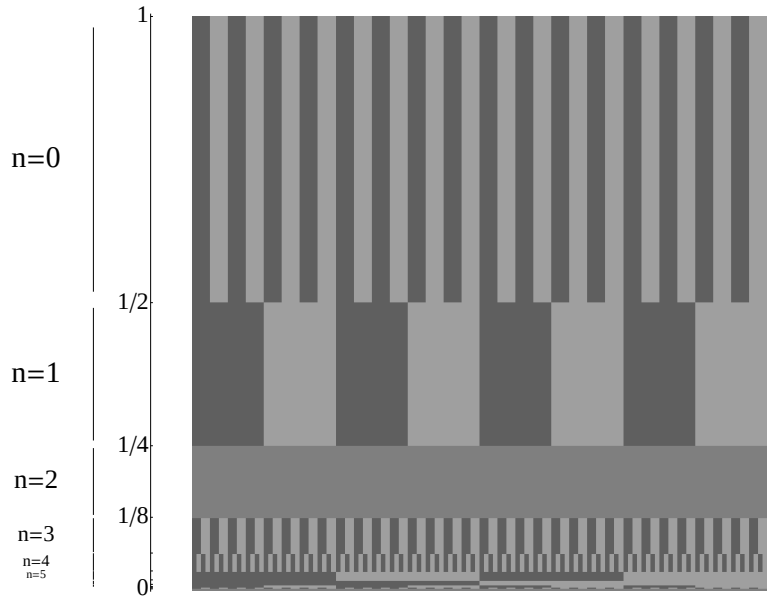

FIGURE 1. A visualization of the joint distribution of (X, Y), where Y is uniformly distributed on $[0, 1]$ and satisfies $\mathsf{N} = \lfloor -\log_2 \mathsf{Y} \rfloor$. (Recall that N is geometrically distributed.) The random variables X and Y take values in the unit interval and so their joint distribution is a distribution on the unit square. The x-axis corresponds to values taken by X, and the y-axis corresponds to values taken by Y. The axes on the left hand side illustrate how an assignment $\mathsf{Y} = y$ translates to an assignment of $\mathsf{N} = n = \lfloor -\log_2 y \rfloor$. Note that this is *not* a plot of the joint density, even though this density exists. Instead, consider that each printed pixel corresponds to a rectangular region (albeit a small one) of the unit square. The gray scale intensity is proportional to the probability assigned to that region. Informally, this representation converges to the joint density as the resolution becomes infinite (i.e., as the pixels become infinitesimally small). Because the joint distribution is computable, we can view it at any finite resolution. The striped pattern is obvious when the corresponding Turing machine halts quickly. However, the limited resolution of the printed figure obscures whether, e.g., the row $n = 2$ is uniformly distributed (because $h(2) = \infty$) or striped with a frequency exceeding our resolution (because $1 \ll h(2) < \infty$). Sampling from a vertical slice of this figure, i.e., sampling Y conditioned on a particular value $\mathsf{X} = x$, requires that we decide whether each pattern is uniform, which is tantamount to solving the halting problem.

for every $\ell \in \{0, 1, \ldots, 2^{k+1}-1\}$. It follows immediately that the density p of $2^{k+1}\mathsf{X}_k$ with respect to Lebesgue measure on $[0, 2^{k+1}]$ is given by

$$p(x) = 2^{-k} \cdot \begin{cases} \frac{2}{3}, & \lfloor x \rfloor \text{ even;} \\ \frac{1}{3}, & \lfloor x \rfloor \text{ odd.} \end{cases} \tag{32}$$

and so the density of X_k is obtained by rescaling. In particular,

$$p_{\mathsf{X}_k}(x) = 2^{k+1} \cdot p(2^{k+1}x), \tag{33}$$

completing the proof. □

As X_k admits a density with respect to Lebesgue measure on $[0, 1]$ for all $k \in \mathbb{N} \cup \{\infty\}$, it follows that the conditional distribution of X given N admits a conditional density (with respect to Lebesgue measure on $[0, 1]$) given by $p_{\mathsf{X}|\mathsf{N}}(x|n) := p_{\mathsf{X}_{h(n)}}(x)$. Each of these densities is continuous and bounded on the nondyadic reals, and so they can be combined to form an $\mathbf{P}_{\mathsf{X}}$-almost continuous version of the conditional distribution.

Lemma III.31. *There is a $\mathbf{P}_{\mathsf{X}}$-almost continuous version of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$.*

Proof. By Bayes' rule (Lemma III.19), the probability kernel κ given by

$$\kappa(x, B) := \frac{\sum_{n \in B} p_{\mathsf{X}|\mathsf{N}}(x|n)\, \mathbf{P}\{\mathsf{N} = n\}}{\sum_{n \in \mathbb{N}} p_{\mathsf{X}|\mathsf{N}}(x|n)\, \mathbf{P}\{\mathsf{N} = n\}} \tag{34}$$

is a version of the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{X}]$. Every nondyadic real $x \in [0, 1]$ is a point of continuity of $p_{\mathsf{X}|\mathsf{N}}$, and so the kernel κ is $\mathbf{P}_{\mathsf{X}}$-almost continuous by Lemma III.21. □

Lemma III.32. *For all $m, n \in \mathbb{N}$ all versions κ of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$, and $\mathbf{P}_{\mathsf{X}}$-almost all x, we have*

$$2^{m-n} \cdot \frac{\kappa(x, \{m\})}{\kappa(x, \{n\})} \in \begin{cases} \{\frac{1}{2}, 1, 2\}, & h(n), h(m) < \infty; \\ \{1\}, & h(n) = h(m) = \infty; \\ \{\frac{2}{3}, \frac{3}{4}, \frac{4}{3}, \frac{3}{2}\}, & \textit{otherwise.} \end{cases}$$

Proof. Let κ be as in Equation (34). Let $m, n \in \mathbb{N}$. Then

$$\begin{aligned} \tau(x) &:= 2^{m-n} \cdot \frac{\kappa(x, \{m\})}{\kappa(x, \{n\})} \\ &= 2^{m-n} \cdot \frac{p_{\mathsf{X}|\mathsf{N}}(x|m)\mathbf{P}\{\mathsf{N} = m\}}{p_{\mathsf{X}|\mathsf{N}}(x|n)\mathbf{P}\{\mathsf{N} = n\}} \\ &= \frac{p_{\mathsf{X}_{h(m)}}(x)}{p_{\mathsf{X}_{h(n)}}(x)}. \end{aligned}$$

For $k < \infty$, $p_{\mathsf{X}_k}(x) \in \{\frac{2}{3}, \frac{4}{3}\}$ for $\mathbf{P}_{\mathsf{X}}$-almost all x. Therefore, for $h(n), h(m) < \infty$, $\tau(x) \in \{\frac{1}{2}, 1, 2\}$ for $\mathbf{P}_{\mathsf{X}}$-almost all x. As $p_{\mathsf{X}_\infty}(x) = 1$ for $\mathbf{P}_{\mathsf{X}}$-almost all x, we have $\tau(x) = 1$ for $\mathbf{P}_{\mathsf{X}}$-almost all x when $h(n) = h(m) = \infty$ and $\tau(x) \in \{\frac{2}{3}, \frac{3}{4}, \frac{4}{3}, \frac{3}{2}\}$ otherwise. □

Let $H = \{n \in \mathbb{N} : h(n) < \infty\}$, i.e., the indices of the Turing machines that halt (on input 0). A classic result in computability theory [Tur36] shows that the halting set H is not computable.

Proposition III.33. *The conditional distribution* $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ *is not computable.*

Proof. Suppose the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ were computable. Let n be the index of some Turing machine that halts (on input 0), i.e., for which $h(n) < \infty$, and consider any $m \in \mathbb{N}$.

Let κ be an arbitrary version of $\mathbf{P}[\mathsf{N}|\mathsf{X}]$, and let R be a $\mathbf{P}_{\mathsf{X}}$-measure one set on which κ is computable. Then the function $\tau(\cdot) := 2^{m-n} \cdot \frac{\kappa(\cdot,\{m\})}{\kappa(\cdot,\{n\})}$ is also computable on R, by Corollary III.10. By Lemma III.32, there is a $\mathbf{P}_{\mathsf{X}}$-measure one subset $D \subseteq R$ on which τ exclusively takes values in the set $T = \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, 1, \frac{4}{3}, \frac{3}{2}, 2\}$.

Although $\mathbf{P}_{\mathsf{X}}$-almost all reals in $[0, 1]$ are in D, any particular real may not be. The following construction can be viewed as an attempt to compute a particular point $d \in D$ at which we can evaluate τ. In fact, we need only a finite approximation to d, because τ is computable on D and T is finite.

For each $t \in T$, let B_t be an ideal ball centered at t of radius less than $\frac{1}{6}$, so that $B_t \cap T = \{t\}$. By Definition II.8, for each $t \in T$, there is a c.e. open set $U_t \subseteq [0, 1]$ such that $\tau^{-1}(B_t) \cap R = U_t \cap R$. Because every open interval has positive $\mathbf{P}_{\mathsf{X}}$-measure, if U_t is nonempty, then $U_t \cap D$ is a positive $\mathbf{P}_{\mathsf{X}}$-measure set whose image is $\{t\}$. Thus, $\mathbf{P}_{\mathsf{X}}$-almost all $x \in U_t \cap R$ satisfy $\tau(x) = t$. As $\bigcup_t U_t$ has $\mathbf{P}_{\mathsf{X}}$-measure one, there is at least one $t \in T$ for which U_t is nonempty. Because each U_t is c.e. open, we can compute the index $\hat{t} \in T$ of some nonempty $U_{\hat{t}}$.

By Lemma III.32 and the fact that $h(n) < \infty$, there are two cases:

(i) $\hat{t} \in \{\frac{1}{2}, 1, 2\}$, implying $h(m) < \infty$, or
(ii) $\hat{t} \in \{\frac{2}{3}, \frac{3}{4}, \frac{4}{3}, \frac{3}{2}\}$, implying $h(m) = \infty$.

Because m was arbitrary, and because the mth Turing machine halts if and only if $h(m) < \infty$, we can use τ to compute the halting set H. Therefore if $\mathbf{P}[\mathsf{X}|\mathsf{N}]$ were computable, then H would be computable, a contradiction. □

Because this proof relativizes, we see that if the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{X}]$ is A-computable for some oracle A, then A computes the halting set H.

Computable operations map computable points to computable points, and so we obtain the following consequence.

Theorem III.34. *The operation* $(\mathsf{X}, \mathsf{Y}) \mapsto \mathbf{P}[\mathsf{Y}|\mathsf{X}]$ *of conditioning a pair of real-valued random variables, even when restricted to pairs for which there exists a* $\mathbf{P}_{\mathsf{X}}$*-almost continuous version of the conditional distribution, is not computable.*

It is natural to ask whether this construction can be extended to produce a pair of computable random variables whose conditional distribution is noncomputable but has an *everywhere continuous* version.

6. Noncomputable continuous conditional distributions

As we saw in Section 4, discontinuity poses a fundamental obstacle to the computability of conditional probabilities. As such, it is natural to ask whether we can construct a pair of random variables (Z, N) that are computable and admit an *everywhere* continuous version of the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$, yet for which every version is noncomputable. In fact, this is possible using a construction similar to that of (X, N) in Section 5.

In particular, if we think of the construction of the kth bit of X as an iterative process, we see that there are two distinct stages. During the first stage, which occurs so long as $k < h(\mathsf{N})$, the bits of X simply mimic those of the uniform random variable V. Then during the second stage, once $k \geq h(\mathsf{N})$, the bits mimic that of $\frac{1}{2}(\mathsf{C} + \mathsf{U})$.

Our construction of Z will differ in the second stage, where the bits of Z will instead mimic those of a random variable S specially designed to smooth out the rough edges caused by the biased coin C. In particular, S will be absolutely continuous and its density will be infinitely differentiable.

We will now make the construction precise. We begin by defining several random variables from which we will construct S.

Lemma III.35. *There is a distribution* F *on* $[0, 1]$ *with the following properties:*

- F *is computable.*
- F *admits a density* p_{F} *with respect to Lebesgue measure (on* $[0, 1]$*) which is infinitely differentiable on all of* $[0, 1]$*.*
- $p_{\mathsf{F}}(0) = \frac{2}{3}$ *and* $p_{\mathsf{F}}(1) = \frac{4}{3}$*.*

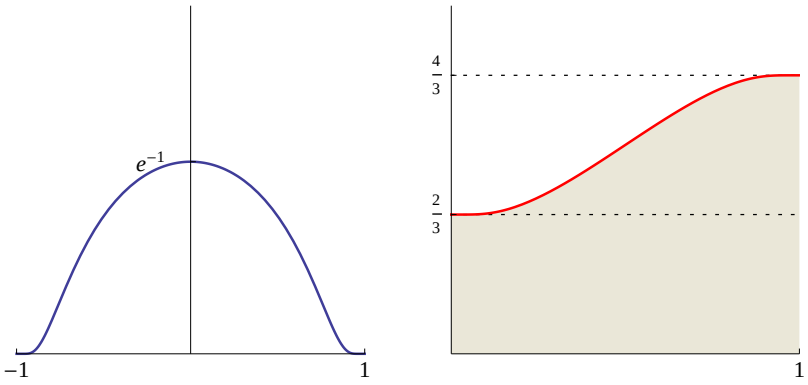

FIGURE 2. (left) $f(x) = e^{-\frac{1}{1-x^2}}$, for $x \in (-1,1)$, and 0 otherwise, a C^∞ bump function whose derivatives at ± 1 are all 0. (right) A density $p(y) = \frac{2}{3}\left(\frac{\Phi(2y-1)}{\Phi(1)} + 1\right)$, for $y \in (0,1)$, of a random variable satisfying Lemma III.35, where $\Phi(y) = \int_{-1}^{y} e^{-\frac{1}{1-x^2}}\,dx$ is the integral of the bump function.

- $\frac{d_+^n}{dx^n} p_{\mathsf{F}}(0) = \frac{d_-^n}{dx^n} p_{\mathsf{F}}(1) = 0$, *for all* $n \geq 1$ *(where* $\frac{d_-^n}{dx^n}$ *and* $\frac{d_+^n}{dx^n}$ *are the left and right derivatives respectively).*

(See Figure 2 for one such random variable.) Note that F is almost surely nondyadic and so the r-th bit F_r of F is a computable random variable.

Let $t \in \{0,1\}^3$. For $r \in \mathbb{N}$, define

$$\mathsf{S}_r^{000} := \begin{cases} 0, & r < 3; \\ \mathsf{F}_{r-3}, & r \geq 3; \end{cases}$$

$$\mathsf{S}_r^{100} := \begin{cases} 1, & r = 0; \\ 0, & 1 \leq r < 3; \\ 1 - F_{r-3}, & r \geq 3; \end{cases}$$

$$\mathsf{S}_r^{t} := \begin{cases} C, & r = 0; \\ t(r), & 1 \leq r < 3; \\ \mathsf{U}_{r-3}, & \text{otherwise}; \end{cases}$$

when $t \notin \{000, 100\}$. It is straightforward to show that S_r^t are computable random variables, uniformly in t and r.

Finally, let T be a uniformly random element in $\{0,1\}^3$, and let the r-th bit of S be $\mathsf{S}_r^{\mathsf{T}}$.

It is straightforward to show that

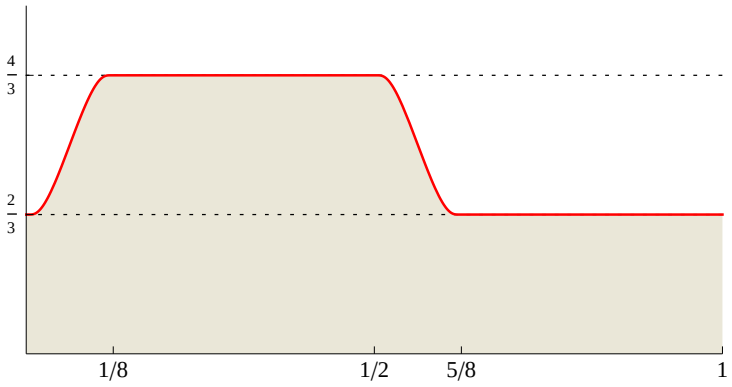

FIGURE 3. Graph of the density function p_{S}.

(i) S admits a density p_{S} with respect to Lebesgue measure on $[0,1]$.

(ii) p_{S} is infinitely differentiable everywhere with $\frac{d^n_+}{dx^n}p_{\mathsf{S}}(0) = \frac{d^n_-}{dx^n}p_{\mathsf{S}}(1)$, for all $n \geq 0$.

(For a visualization of the density p_{S} see Figure 3.)

We say a real $x \in [0,1]$ is **valid for** S if $x \in (\frac{1}{8}, \frac{4}{8}) \cup (\frac{5}{8}, \frac{8}{8})$. (For nondyadic x, this is equivalent to the first 3 bits of the binary expansion of x not being 000 or 100.) The following are then straightforward consequences of the construction of S and the definition of valid points:

(iii) If x is valid for S then $p_{\mathsf{S}}(x) \in \{\frac{2}{3}, \frac{4}{3}\}$.

(iv) The Lebesgue measure (and $\mathbf{P}_{\mathsf{S}}$-measure) of the collection of valid x is $\frac{3}{4}$.

Next we define, for every $k \in \mathbb{N}$, the random variables Z_k mimicking the construction of X_k. Specifically, for $k \in \mathbb{N}$, define

$$\mathsf{Z}_k := \frac{\lfloor 2^k \mathsf{V} \rfloor + \mathsf{S}}{2^k}, \tag{35}$$

and let $\mathsf{Z}_\infty := \lim_{k\to\infty} \mathsf{Z}_k = \mathsf{V}$. Then the nth bit of Z_k is

$$(\mathsf{Z}_k)_n = \begin{cases} \mathsf{V}_n, & n < k; \\ \mathsf{S}_{n-k}, & n \geq k \end{cases} \qquad \text{a.s.} \tag{36}$$

For $k < \infty$, we say that $x \in [0,1]$ is **valid for** Z_k if the fractional part of $2^k x$ is valid for S, and we say that x is **valid for** Z_∞ for all x. Let A_k be the collection of x valid for Z_k. It follows from *(iv)* that the Lebesgue measure of A_k is $\frac{3}{4}$ for all $k < \infty$.

It is straightforward to show from *(i)* and *(ii)* above that Z_k admits a density p_{Z_k} with respect to Lebesgue measure on $[0,1]$ and that this density is infinitely differentiable.

To complete the construction, we define $\mathsf{Z} := \mathsf{Z}_{h(\mathsf{N})}$. The following results are analogous to those in the almost continuous construction:

Lemma III.36. *The random variable* Z *is computable.*

Lemma III.37. *There is an everywhere continuous version of* $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$.

Proof. The density p_{Z} is everywhere continuous and positive. □

Lemma III.38. *For all* $m, n \in \mathbb{N}$*, all version* κ *of the conditional distribution* $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$ *and* $\mathbf{P}_{\mathsf{Z}}$*-almost all* x*, if* x *is valid for* $\mathsf{Z}_{h(n)}$ *and for* $\mathsf{Z}_{h(m)}$ *then*

$$2^{m-n} \cdot \frac{\kappa(x, \{m\})}{\kappa(x, \{n\})} \in \begin{cases} \{\frac{1}{2}, 1, 2\}, & h(n), h(m) < \infty; \\ \{1\}, & h(n) = h(m) = \infty; \\ \{\frac{2}{3}, \frac{3}{4}, \frac{4}{3}, \frac{3}{2}\}, & \textit{otherwise.} \end{cases}$$

We now show that one can compute the halting set from any version of the conditional distribution.

Proposition III.39. *The conditional distribution* $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$ *is not computable.*

Proof. Suppose the conditional distribution $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$ were computable. Let n be the index of some Turing machine that does not halt (on input 0), i.e., for which $h(n) = \infty$. Consider any $m \in \mathbb{N}$. Notice that all $x \in [0, 1]$ are valid for $\mathsf{Z}_{h(n)}$ and so $A_{h(n)} \cap A_{h(m)} = A_{h(m)}$.

Let κ be an arbitrary version of $\mathbf{P}[\mathsf{N}|\mathsf{Z}]$, and let R be a $\mathbf{P}_{\mathsf{Z}}$-measure one set on which κ is computable. Then the function

$$\tau(\cdot) := 2^{m-n} \cdot \frac{\kappa(\cdot, \{m\})}{\kappa(\cdot, \{n\})} \tag{37}$$

is also computable on R. Define $T_\infty := \{1\}$, $T_{<\infty} := \{\frac{2}{3}, \frac{4}{3}\}$ and $T := T_\infty \cup T_{<\infty}$.

By equation (37), there is a $\mathbf{P}_{\mathsf{Z}}$-measure one subset $D \subseteq R$ such that whenever $x \in D \cap A_{h(m)}$ then $\tau(x)$ is in T.

For $t \in T$, let B_t be an ideal ball of radius less than $\frac{1}{6}$ about t, and let U_t be a c.e. open set such that $\tau^{-1}(B_t) \cap R = U_t \cap R$. Define $U_\infty := U_1$ and $U_{<\infty} := U_{\frac{2}{3}} \cup U_{\frac{4}{3}}$. Notice these are both c.e. open sets and $D \cap U_\infty \cap U_{<\infty} = \emptyset$.

We now consider two cases. First, assume $h(m) = \infty$. In this case $A_{h(m)} = [0, 1]$ and $A_{h(m)} \cap D \subseteq \tau^{-1}(T_\infty) \cap D = U_\infty \cap D$. Hence

(a) The Lebesgue measure of U_∞ is $1 > \frac{1}{2}$.

If, however, $h(m) < \infty$ then $A_{h(m)}$ has Lebesgue measure $\frac{3}{4}$ and $A_{h(m)} \subseteq \tau^{-1}(T_{<\infty}) \cap D = U_{<\infty} \cap D$. So

(b) The Lebesgue measure of $U_{<\infty}$ is at least $\frac{3}{4} > \frac{1}{2}$.

In particular, for each $m \in \mathbb{N}$ exactly one of *(a)* or *(b)* must hold. But it is clear that the collection of m for which *(b)* holds a c.e. set and the collection of m for which *(b)* does not hold (i.e., for which *(a)* holds) is also a c.e. set. So, as *(b)* holds of m if and only if $m \in H = \{m : h(m) < \infty\}$, we have H is a computable set, which we know is a contradiction.

Therefore κ must not be computable. □

In conclusion, we obtain the following strengthening of Theorem III.34.

Theorem III.40. *Let* X *and* Y *be computable real-valued random variables. Then operation* $\mathsf{X}, \mathsf{Y} \mapsto \mathbf{P}[\mathsf{X}|\mathsf{Y}]$ *of conditioning a pair of real-valued random variables, even when restricted to pairs for which there exists an everywhere continuous version of the conditional distribution, is not computable.*

Despite these fundamental noncomputability results, many important questions remain: How badly noncomputable is conditioning, even restricted to these continuous settings? What is the computational *complexity* of conditioning on *efficiently* computable continuous random variables? In what restricted settings is conditioning *computable*? In the final section, we begin to address the latter of these.

7. Conditioning is Turing jump computable

Having demonstrated a pair of computable random variables for which conditioning is at least as hard as the halting problem, we now show that it is no harder. Recall that the **Turing jump** x' of a real $x \in \mathbb{R}$ is given by the halting set relative to the oracle x. (In particular, the halting set H is Turing equivalent to $0'$.)

Theorem III.41. *Let* X *be a computable random variable in* $\mathbb{R}$*, and let* Y *be a computable random variable in a computable metric space* T*. If* $x \in \mathbb{R}$ *is a point with the Tjur property (for* Y *given* X*) then* $\mathbf{P}^x_{\mathsf{Y}}$ *is* x'*-computable.*

Proof. Begin by x-computing a nested sequence $\{V_i\}_{i\in\mathbb{N}}$ of rational intervals that converge rapidly to x, i.e., $V_{i+1} \subseteq V_i$ and $\bigcap_i V_i = \{x\}$, and V_i is contained in the 2^{-i}-ball around x. Because x has the Tjur property, x is in the support of X. Hence, for each $i \in \mathbb{N}$, we have $\mathbf{P}_{\mathsf{X}}(V_i) > 0$. Hence, by Lemma II.22 and Remark II.24, for each i we can compute a $\mathbf{P}_{\mathsf{X}}$-almost decidable set $B_i \subseteq V_i$ such that $\mathbf{P}_{\mathsf{X}}(B_i) > 0$. Note that, by Lemma III.13, the sequence $\{\mathbf{P}_{\mathsf{Y}}(\cdot|B_i)\}$ is a computable sequence of computable measures, uniformly in i.

By construction, the sequence $\{(V_i, B_i)\}_{i\geq 1}$ is cofinal in $\mathscr{D}(x)$, and so by the Tjur property of x,

$$\mathbf{P}^x_{\mathsf{Y}}(A) = \lim_{i\to\infty} \mathbf{P}(\mathsf{Y} \in A | \mathsf{X} \in B_i), \tag{38}$$

for every measurable set A.

Let A be an arbitrary $\mathbf{P}_{\mathsf{Y}}$-almost decidable set. For each i, by Lemma II.21, $\mathbf{P}(\mathsf{Y} \in A|\mathsf{X} \in B_i)$ is a computable real. Hence the sequence $\{\mathbf{P}(\mathsf{Y} \in A|\mathsf{X} \in B_i)\}_{i\geq 1}$ is an x-computable sequence of reals, uniformly in A. By Eq. (38) and the limit lemma for reals (see, e.g., [Zhe02, Thm. 9.4]), $\mathbf{P}^x_{\mathsf{Y}}(A)$ is an x'-computable real, uniformly in A. Hence, by Corollary II.23, $\mathbf{P}^x_{\mathsf{Y}}$ is x'-computable. □

8. Continuity in the setting of identifiability in the limit

Osherson, Stob, and Weinstein [OSW88] study learning theory in the setting of identifiability in the limit (see [Gol67] and [Put85] for more details on this setting) and prove that a so-called "computable Bayesian" learner fails to identify the index of a c.e. set that is otherwise computably identifiable in the limit. From the perspective of computable analysis, this work can be interpreted as studying when certain conditional distributions are $0'$-computable, rather than computable. A close analysis of their construction reveals that the conditional distribution they define is an everywhere discontinuous function, hence fundamentally not computable in the same way as our much more elementary construction involving a mixture of two measures concentrated on the rationals and irrationals, respectively (see Proposition III.28). As we have argued, the more appropriate operator to study is that restricted to those random variables whose conditional distributions admit almost continuous versions. For the interested reader, we now give a close analysis of the construction of Osherson, Stob, and Weinstein from the perspective of the computability of conditional distributions.

Let $g : \mathbb{N}^* \to \mathbb{N}$ be a "learner", i.e., a map taking finite prefixes of integers (of the infinite integer "text" we are trying to "identify") to integers (guesses at an index for a c.e. set containing exactly the numbers in the infinite text). Assuming that the index and sequence are random, a learner g is "Bayesian" if $g(\bar{t}_n)$ is the most likely index given the evidence "$\bar{t}_n$ is a finite prefix of the infinite sequence t".

[OSW88] give a computable learner g that is Bayesian. But it does not "solve" the inference problem, because for a positive measure set of strings, the limit

$$\lim_n g(t_n) \tag{39}$$

does not exist. In this case, both values that it oscillates between are "correct" answers, in the sense that they both code the same (c.e.) set. In order for a learner to recognize when this oscillation is unnecessary, Osherson, Stob, and Weinstein show that it is necessary and sufficient to decide whether a certain program has a domain greater than 2. This property is only semidecidable, and so no computable Bayesian learner exists.

One might object that oscillating between two correct answers is satisfactory, however, this property of "confirmation" is a fundamental one in the philosophical foundations of the learning in the limit framework. However, the authors point out at the end of the paper that the result relies on this particular definition of learnability and that other definitions may see the result change. In fact, in [OSW86] they describe more general settings that include the so-called "intensional" setting. In this setting, the computable Bayesian learner they describe would be considered to solve the inductive inference problem.[2]

How does their construction relate to the problem of computing conditional distributions or probabilities? Below, we show that the conditional distribution driving the (maximum *a posteriori*) decisions of the "Bayesian learner" is discontinuous on every measure one set. We have already argued that a more interesting setting is one in which we are guaranteed the existence of an almost continuous version. However, it seems likely that other results in the identifiability in the limit literature might bear on the computability of important operations in statistics.

We now proceed to study the aforementioned conditional distribution in detail. [OSW86] fix four enumerations of the natural numbers E_i, $i = 1, \ldots, 4$, and distributions $\{p_n : n \in \mathbb{N}\}$ and then define the following random variables:

(1) A geometric random variable N and discrete random variable D, uniformly distributed on $\{1, 2, 3, 4\}$;
(2) A derived random variable $\mathsf{K} := E_{\mathsf{D}}(\mathsf{N})$, corresponding to the N'th entry in the D'th enumeration;
(3) An i.i.d. sequence T of draws from p_{K}.

The random index K encodes, among other things, a distribution p_{K} on the symbols $\{A, B, C, \langle 0\rangle, \langle 1\rangle, \langle 2\rangle, \ldots\}$. In fact, K determines this distribution by way of coding an enumeration of the support $\mathscr{S}(\mathsf{K})$ of p_{K}. Roughly, the k'th entry in this enumeration is returned with probability 2^{-k-1} (the enumeration may repeat itself and in this way

[2]Historically, the "identifiabilty in the limit" framework was eventually supplanted by the vastly more productive "probably approximately correct", or PAC, framework introduced by Valiant [Val84].

can give some value with nondyadic probability). Let $\mathrm{rng}(\mathsf{K}) = \{J : \mathscr{S}(J) = \mathscr{S}(\mathsf{K})\}$. Roughly any answer in $\mathrm{rng}(\mathsf{K})$ counts as a correct answer when identifying a set in the limit.

The "inductive inference problem" that [OSW86] put forth is, given T, return an integer in $\mathrm{rng}(\mathsf{K})$ (i.e., any code for $\mathscr{S}(\mathsf{K})$). In their setting, a learner receives a finite prefix and returns a guess. The Bayesian is forced to return a guess k such that $\mathbf{P}[k \in \mathrm{rng}(\mathsf{K})|\mathsf{T}_n]$ is maximized.

We will now characterize $\mathbf{P}[\mathscr{S}(\mathsf{K})|\mathsf{T}]$.

By construction of the distribution (see [OSW88] for details), one learns the value N from observing an almost surely finite prefix of T. Let $\eta : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ be a map such that

$$\eta(\mathsf{T}) = \mathsf{N} \quad \text{a.s.} \tag{40}$$

That is, η is the function that (almost always) reads off the value of N from T. Given the details in [OSW88], it is easy to see that η is almost computable (and hence, almost continuous).

Furthermore, by construction, with probability one,

$$\mathscr{S}(E_1(\mathsf{N})) = \mathscr{S}(E_2(\mathsf{N})) \tag{41}$$

and

$$\mathscr{S}(E_3(\mathsf{N})) = \mathscr{S}(E_4(\mathsf{N})), \tag{42}$$

and in fact, sometimes

$$\mathscr{S}(E_1(\mathsf{N})) = \mathscr{S}(E_3(\mathsf{N})). \tag{43}$$

More precisely, $\mathscr{S}(E_1(\mathsf{N})) \subseteq \mathscr{S}(E_3(\mathsf{N}))$ a.s., and

$$\mathscr{S}(E_3(\mathsf{N})) \setminus \mathscr{S}(E_2(\mathsf{N})) \subseteq \{C\}, \tag{44}$$

i.e., when they differ, the former simply includes the extra symbol C. To describe when (43) holds, define $D(n)$ to be the cardinality of the domain of the n'th program. By construction, $\mathscr{S}(E_1(n)) = \mathscr{S}(E_3(n))$ if and only if $D(n) \leq 2$.

When $D(\mathsf{N}) > 2$ and $C \in \mathscr{S}(\mathsf{K})$, then C will appear in T almost surely (in fact it will appear an infinite number of times). Therefore, a version κ of $\mathbf{P}[\mathscr{S}(\mathsf{K})|\mathsf{T}]$ is a random delta distribution

$$\kappa(t, \cdot) = \delta_{M(t)}(\cdot) \tag{45}$$

whose location $M(t)$ is a.e. unique and given by

$$M(t) = \begin{cases} \mathscr{S}(E_1(n)) & D(n) \leq 2 \text{ or } t \text{ does not contain a } C \\ \mathscr{S}(E_3(n)) & \text{otherwise,} \end{cases} \tag{46}$$

where $n = \eta(t)$. We now discuss the continuity of versions of $\mathbf{P}[\mathscr{S}(\mathsf{K})|\mathsf{T}]$, or equivalently, M. The set

$$\Delta := \{t \in \mathbb{N}^{\mathbb{N}} | D(\eta(t)) \leq 2\} \tag{47}$$

is an (almost) open set in Baire space because η is almost continuous and indicator functions continuous on $\mathbb{N}$ (though not computable, in this case). Note that from (46), we see that, restricted to the set $\{\mathsf{T} \in \Delta\}$,

$$\mathscr{S}(\mathsf{K}) = \mathscr{S}(E_1(\mathsf{N})) \quad \text{a.s.} \tag{48}$$

and so there exists a *computable on* $\{\mathsf{T} \in \Delta\}$ version of M. We now consider the complement, $\{\mathsf{T} \notin \Delta\}$. In this case, M simplifies to

$$M(t) = \begin{cases} \mathscr{S}(E_1(n)) & t \text{ does not contain a } C \\ \mathscr{S}(E_3(n)) & \text{otherwise,} \end{cases} \tag{49}$$

with $n = \eta(t)$. The set

$$\{t | t \text{ contains at least one C}\} \tag{50}$$
$$= \bigcup \{t \times \mathbb{N}^{\mathbb{N}} | t \text{ is a finite string of integers ending with a C}\}$$

is open in Baire space, but not clopen. This implies that M is discontinuous (as well as every restriction of M to a measure one set). To see this, assume otherwise, and let t be such that $M(t) = \mathscr{S}(E_1(\eta(t)))$. Then

$$M(\bar{t}_m) = \{\mathscr{S}(E_1(\eta(t))\} \tag{51}$$

for some m-length prefix $\bar{t}_m$ of t. But, of course, $\bar{t}_m$, being an open set, contains strings with C's in them, so M is discontinuous at t. As t was arbitrary, it follows that M is discontinuous everywhere. Showing that M remains discontinuous after restricting its domain to a measure one subset is straightforward.

To connect this with the question of identifiability in the limit, note that, because the conditional distribution is a delta distribution, finding the code k such that

$$\mathbf{P}[k \in \text{rng}(\mathsf{K})|\mathsf{T}] \geq \mathbf{P}[j \in \text{rng}(\mathsf{K})|\mathsf{T}] \tag{52}$$

for all j is tantamount to computing $\mathbf{P}[\mathscr{S}(\mathsf{K})|\mathsf{T}]$. It is then easy to see that any version of the conditional probability on the left is not almost continuous.

We now turn to the study of exchangeable sequences of random variables.

CHAPTER IV

Exchangeable sequences and de Finetti's theorem

This chapter examines the computable probability theory of exchangeable sequences of real-valued random variables. The notion of *exchangeability*, that the probability distribution of (i.e., our prior knowledge about) a sequence of data $X_1, X_2, \ldots$ does not depend on the ordering of the data, plays a central role in hierarchical Bayesian modeling [BS94]. The classical de Finetti theorem states that an exchangeable sequence of real random variables is a mixture of independent and identically distributed (i.i.d.) sequences of random variables. Moreover, there is an (almost surely unique) measure-valued random variable, called the *directing random measure*, conditioned on which the random sequence is i.i.d. The distribution of the directing random measure is called the *de Finetti measure* or the *mixing measure*.

In statistics and machine learning, we often have a description of an exchangeable sequence $X_1, X_2, \ldots$ in terms of an *algorithm which samples* the elements in order. As an example, two of the most famous stochastic processes in Bayesian nonparametrics — the Chinese Restaurant process and the Indian Buffet process — have such descriptions. This chapter studies the following question: What can an *algorithmic* description tell us about the directing random measure? In particular, given an algorithm for sampling the exchangeable sequence, is there an algorithm for sampling the directing random measure? And if so, how can we find it?

As an example, consider the Indian Buffet process, which has been applied to the problem of modeling visual scenes, analogical reasoning and many other phenomena. While its de Finetti measure has been identified [TJ07, TGG07], no exact algorithm is known for generating independent samples: both the inverse Levy measure [WI98] and stick breaking constructions [TGG07] fall short of producing exact samples. A new representation would suggest new algorithms for this important modeling tool.

We prove a computable version of de Finetti's theorem: the distribution of an exchangeable sequence of real random variables is computable if and only its de Finetti measure is computable. The classical proofs

are non-constructive and do not immediately suggest a computable approach; instead, we show how to directly compute the de Finetti measure (as characterized by the classical theorem) in terms of a computable representation of the distribution of the exchangeable sequence. Along the way, we prove that a distribution on $[0,1]^\omega$ is computable if and only if its moments are uniformly computable, which may be of independent interest. The result has some immediate corollaries including one that explains exactly when we should expect to be able to predict the directing random measure from observations of the corresponding exchangeable sequence [FR10], a question that we explore in Section 6.

A key step in the proof is to describe the de Finetti measure in terms of the moments of a set of random variables derived from the exchangeable sequence. When the directing random measure is (almost surely) continuous, we can show that these moments are computable, which suffices to complete the proof of the main theorem in this case. In the general case, we give a proof inspired by a randomized algorithm that, with probability one, computes the de Finetti measure.

Exchangeable sequences play a fundamental role in both statistical models and their implementation on computers. Given a *sequential* description of an exchangeable process, in which one uses previous samples or sufficient statistics to sample the next element in the sequence, a direct implementation in these languages would need to use non-local communication (to access old samples or update sufficient statistics). This is often implemented by modifying the program's internal state directly (i.e., using *mutation*), or via some indirect method such as a state monad. The classical de Finetti theorem implies that (for such sequences over the reals) there is an alternative description in which samples are conditionally independent (and so could be implemented without non-local communication), thereby allowing parallel implementations. But the classical result does not imply that there is a *program* that computes the sequence according to this description. Even when there is such a program, the classical theorem does not provide a method for finding it. The computable de Finetti theorem states that such a program *does* exist. Moreover, the proof itself provides the *method* for constructing the desired program. In Section 5 we describe how an implementation of the computable de Finetti theorem performs a code transformation that eliminates the use of non-local state in procedures that induce exchangeable stochastic processes.

This transformation is of interest beyond its implications for programming language semantics. In statistics and machine learning, it is often desirable to know the representation of an exchangeable stochastic process in terms of its de Finetti measure (for several examples,

see Section 5.3). Many such processes in machine learning have very complicated (though computable) distributions, and it is not always feasible to find the de Finetti representation by hand. The computable de Finetti theorem provides a method for automatically obtaining such representations.

1. de Finetti's Theorem

Fix a basic probability space $(\Omega, \mathcal{F}, \mathbf{P})$ and let $\mathcal{B}_{\mathbb{R}}$ denote the Borel sets of $\mathbb{R}$. Note that we will use ω to denote the set of nonnegative integers (as in logic), rather than an element of the basic probability space Ω (as in probability theory). By a *random measure* we mean a random element in the space of Borel measures on $\mathbb{R}$, i.e., a kernel from $(\Omega, \mathcal{F})$ to $(\mathbb{R}, \mathcal{B}_{\mathbb{R}})$. An event $A \in \mathcal{F}$ is said to occur *almost surely* (a.s.) if $\mathbf{P}A = 1$. We denote the indicator function of a set B by $\mathbf{1}_B$. Unlike previous chapters, we will not use a sans serif font for random variables.

Definition IV.1 (Exchangeable sequence)**.** Let $X = \{X_i\}_{i\geq 1}$ be a sequence of real random variables. We say that X is *exchangeable* if, for every finite set $\{k_1, \ldots, k_j\}$ of distinct indices, $(X_{k_1}, \ldots, X_{k_j})$ is equal in distribution to $(X_1, \ldots, X_j)$.

Theorem IV.2 (de Finetti [Kal05, Chap. 1.1])**.** *Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence of real-valued random variables. There is a random probability measure ν on $\mathbb{R}$ such that $\{X_i\}_{i\geq 1}$ is conditionally i.i.d. with respect to ν. That is,*

$$\mathbf{P}[X \in \cdot \mid \nu] = \nu^{\infty} \quad \text{a.s.} \tag{53}$$

Moreover, ν is a.s. unique and given by

$$\nu(B) = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_B(X_i) \quad \text{a.s.}, \tag{54}$$

where B ranges over $\mathcal{B}_{\mathbb{R}}$.

The random measure ν is called the *directing random measure*.[1] Its distribution (a measure on probability measures), which we denote by μ, is called the *de Finetti measure* or the *mixing measure*. As in Kallenberg [Kal05, Chap. 1, Eq. 3], we may take expectations on both sides of (53) to arrive at a characterization

$$\mathbf{P}\{X \in \cdot\} = \mathbb{E}\nu^{\infty} = \int m^{\infty} \mu(dm) \tag{55}$$

[1] The directing random measure is only unique up to a null set, but it is customary to refer to it as if it were unique, as long as we only rely on almost-sure properties.

of an exchangeable sequence as a mixture of i.i.d. sequences.

A Bayesian perspective suggests the following interpretation: exchangeable sequences arise from independent observations from a latent measure ν. Posterior analysis follows from placing a prior distribution on ν. For further discussion of the implications of de Finetti's theorem for the foundations of statistical inference, see Dawid [Daw82] and Lauritzen [Lau84].

In 1931, de Finetti [dF31] proved the classical result for binary exchangeable sequences, in which case the de Finetti measure is simply a mixture of Bernoulli distributions; the exchangeable sequence is equivalent to repeatedly flipping a coin whose weight is drawn from some distribution on $[0, 1]$. In 1937, de Finetti [dF37] extended the result to arbitrary real-valued exchangeable sequences. We will refer to this more general version as the *de Finetti theorem*. Later, Hewitt and Savage [HS55] extended the result to compact Hausdorff spaces, and Ryll-Nardzewski [RN57] introduced a weaker notion than exchangeability that suffices to give a conditionally i.i.d. representation. Hewitt and Savage [HS55] provide a history of the early developments, and a discussion of some subsequent extensions can be found in Kingman [Kin78], Diaconis and Freedman [DF84], and Aldous [Ald85]. A recent book by Kallenberg [Kal05] provides a comprehensive view of the area of probability theory that has grown out of de Finetti's theorem, stressing the role of invariance under symmetries.

1.1. Examples. Consider an exchangeable sequence of $[0, 1]$-valued random variables. In this case, the de Finetti measure is a distribution on the (Borel) measures on $[0, 1]$. For example, if the de Finetti measure is a Dirac measure on the uniform distribution on $[0, 1]$ (i.e., the distribution of a random measure which is almost surely the uniform distribution), then the induced exchangeable sequence consists of independent, uniformly distributed random variables on $[0, 1]$.

As another example, let p be a random variable, uniformly distributed on $[0, 1]$, and let $\nu := \delta_p$, i.e., the Dirac measure concentrated on p. Then the de Finetti measure is the uniform distribution on Dirac measures on $[0, 1]$, and the corresponding exchangeable sequence is $p, p, \ldots$, i.e., a constant sequence, marginally uniformly distributed.

As a further example, we consider a stochastic process $\{X_i\}_{i\geq 1}$ composed of binary random variables whose finite marginals are given

by

$$\mathbf{P}\{X_1 = x_1, \ldots, X_n = x_n\} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(\alpha+S_n)\Gamma(\beta+(n-S_n))}{\Gamma(\alpha+\beta+n)}, \quad (56)$$

where $S_n := \sum_{i \leq n} x_i$, and where Γ is the Gamma function and α, β are positive real numbers. (One can verify that these marginals satisfy Kolmogorov's extension theorem [Kal02, Theorem 6.16], and so there is a stochastic process $\{X_i\}_{i \geq 1}$ with these finite marginals.) Clearly this process is exchangeable, as n and S_n are invariant to order. This process can also be described by a sequential scheme known as Pólya's urn [dF75, Chap. 11.4]. Each X_i is sampled in turn according to the conditional distribution

$$\mathbf{P}\{X_{n+1} = 1 \mid X_1 = x_1, \ldots, X_n = x_n\} = \frac{\alpha + S_n}{\alpha + \beta + n}. \quad (57)$$

This process is often described as repeated sampling from an urn: starting with α red balls and β black balls, a ball is drawn at each stage uniformly at random, and then returned to the urn along with an additional ball of the same color. By de Finetti's theorem, there exists a random variable $\theta \in [0, 1]$ with respect to which the sequence is conditionally independent and $\mathbf{P}\{X_i = 1 \mid \theta\} = \theta$ for each i. In fact,

$$\mathbf{P}[X_1 = x_1, \ldots, X_n = x_n \mid \theta\,] = \prod_{i \leq n} \mathbf{P}[X_i = x_i \mid \theta\,] = \theta^{S_n}(1-\theta)^{(n-S_n)}. \quad (58)$$

Furthermore, one can show that θ is Beta(α, β)-distributed, and so the process given by the marginals (56) is called the Beta-Bernoulli process. Finally, the de Finetti measure is the distribution of the random Bernoulli measure $\theta\delta_1 + (1-\theta)\delta_0$.

1.2. The Computable de Finetti Theorem. In each of these examples, the de Finetti measure is a *computable measure*. (In Section 2, we make this and related notions precise. For an implementation of the Beta-Bernoulli process in a probabilistic programming language, see in Section 5.) A natural question to ask is whether computable exchangeable sequences always arise from computable de Finetti measures. In fact, computable de Finetti measures give rise to computable distributions on exchangeable sequences (see Proposition IV.17). Our main result is the converse: every computable distribution on real-valued exchangeable sequences arises from a computable de Finetti measure.

Theorem IV.3 (Computable de Finetti)**.** *Let χ be the distribution of a real-valued exchangeable sequence X, and let μ be the distribution of its directing random measure ν. Then μ is computable relative to χ, and χ is computable relative to μ. In particular, χ is computable if and only if μ is computable.*

The directing random measure is classically given a.s. by the explicit limiting expression (54). Without a computable handle on the rate of convergence, the limit is not directly computable, and so we cannot use this limit directly to compute the de Finetti measure. However, we are able to reconstruct the de Finetti measure using the moments of random variables derived from the directing random measure.

Outline of the Proof. Recall that $\mathcal{B}_{\mathbb{R}}$ denotes the Borel sets of $\mathbb{R}$. Let $\mathcal{I}_{\mathbb{R}}$ denote the set of open intervals, and let $\mathcal{I}_{\mathbb{Q}}$ denote the set of open intervals with rational endpoints. Then $\mathcal{I}_{\mathbb{Q}} \subsetneq \mathcal{I}_{\mathbb{R}} \subsetneq \mathcal{B}_{\mathbb{R}}$. For $k \geq 1$ and $\boldsymbol{\beta} \in \mathcal{B}_{\mathbb{R}}^k = \mathcal{B}_{\mathbb{R}} \times \cdots \times \mathcal{B}_{\mathbb{R}}$, we write $\boldsymbol{\beta}(i)$ to denote the ith coordinate of $\boldsymbol{\beta}$.

Let $X = \{X_i\}_{i \geq 1}$ be an exchangeable sequence of real random variables, with distribution χ and directing random measure ν. For every $\boldsymbol{\gamma} \in \mathcal{B}_{\mathbb{R}}$, we define a $[0,1]$-valued random variable $V_{\boldsymbol{\gamma}} := \nu\boldsymbol{\gamma}$. A classical result in probability theory [Kal02, Lem. 1.17] implies that a Borel measure on $\mathbb{R}$ is uniquely characterized by the mass it places on the open intervals with rational endpoints. Therefore, the distribution of the stochastic process $\{V_\tau\}_{\tau \in \mathcal{I}_{\mathbb{Q}}}$ determines the de Finetti measure μ (the distribution of ν).

Definition IV.4 (Mixed moments)**.** Let $\{x_i\}_{i \in C}$ be a family of random variables indexed by the set C. The *mixed moments of* $\{x_i\}_{i \in C}$ are the expectations $\mathbb{E}\big(\prod_{i=1}^k x_{j(i)}\big)$, for $k \geq 1$ and $j \in C^k$.

We can now restate the consequence of de Finetti's theorem described in Eq. (55), in terms of the finite-dimensional marginals of the exchangeable sequence X and the mixed moments of $\{V_{\boldsymbol{\beta}}\}_{\boldsymbol{\beta} \in \mathcal{B}_{\mathbb{R}}}$.

Corollary IV.5. $\mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \boldsymbol{\beta}(i)\}\big) = \mathbb{E}\big(\prod_{i=1}^k V_{\boldsymbol{\beta}(i)}\big)$ *for $k \geq 1$ and $\boldsymbol{\beta} \in \mathcal{B}_{\mathbb{R}}^k$.*

For $k \geq 1$, let $\mathcal{L}_{\mathbb{R}^k}$ denote the set of finite unions of open rectangles in $\mathbb{R}^k$ (i.e., the lattice generated by $\mathcal{I}_{\mathbb{R}}^k$), and let $\mathcal{L}_{\mathbb{Q}^k}$ denote the set of finite unions of open rectangles in $\mathbb{Q}^k$. (Note that $\mathcal{I}_{\mathbb{Q}} \subsetneq \mathcal{L}_{\mathbb{Q}} \subsetneq \mathcal{L}_{\mathbb{R}} \subsetneq \mathcal{B}_{\mathbb{R}}$.) As we will show in Lemma IV.10, when χ is computable, we can enumerate all rational lower bounds on quantities of the form

$$\mathbf{P}\big(\textstyle\bigcap_{i=1}^k \{X_i \in \sigma(i)\}\big), \tag{59}$$

where $k \geq 1$ and $\sigma \in \mathcal{L}_{\mathbb{Q}}^k$.

In general, we cannot enumerate all rational upper bounds on (59). However, if $\sigma \in \mathcal{L}_{\mathbb{Q}}^k$ (for $k \geq 1$) is such that, with probability one, ν places no mass on the boundary of any $\sigma(i)$, then $\mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \sigma(i)\}\big) = \mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \overline{\sigma(i)}\}\big)$, where $\overline{\sigma(i)}$ denotes the closure of $\sigma(i)$. In this case, for every rational upper bound q on (59), we have that $1 - q$ is a lower bound on

$$\mathbf{P}\big(\bigcup_{i=1}^k \{X_i \notin \overline{\sigma(i)}\}\big), \tag{60}$$

a quantity for which we can enumerate all rational lower bounds. If this property holds for all $\sigma \in \mathcal{L}_{\mathbb{Q}}^k$, then we can compute the mixed moments $\{V_\tau\}_{\tau \in \mathcal{L}_{\mathbb{Q}}}$. A natural condition that implies this property for all $\sigma \in \mathcal{L}_{\mathbb{Q}}^k$ is that ν is a.s. continuous (i.e., with probability one, $\nu\{x\} = 0$ for every $x \in \mathbb{R}$).

In Section 3, we show how to computably recover a distribution from its moments. This suffices to recover the de Finetti measure when ν is a.s. continuous, as we show in Section 4.1. In the general case, point masses in ν can prevent us from computing the mixed moments. Here we use a proof inspired by a randomized algorithm that almost surely avoids the point masses and recovers the de Finetti measure. For the complete proof, see Section 4.3.

2. Computable Representations

In this chapter, we take a topological perspective on computability. For metrizable spaces, these notions align with those described in Chapter II. We begin by introducing notions of computability on various topological spaces. These definitions follow from more general notions studied in the framework of Type-2 effectivity [Wei00b], although we will sometimes derive simpler equivalent representations for the concrete spaces we need (such as the real numbers, Borel measures on reals, and Borel measures on Borel measures on reals). For details, see the original papers, as noted.

We assume familiarity with standard notions of computability as briefly outlined in Chapter II. Recall that $r \in \mathbb{R}$ is a *c.e. real* (sometimes called a *left-c.e. real*) when the set of all rationals less than r is a c.e. set. Similarly, r is a *co-c.e. real* (sometimes called a *right-c.e. real*) when the set of all rationals greater than r is c.e. A real r is a computable real when it is both a c.e. and co-c.e. real.

To represent more general spaces, we work in terms of an effectively presented topology. Suppose that S is a second-countable T_0 topological space with subbasis $\mathcal{S}$. For every point $x \in S$, define the set $\mathcal{S}_x :=$

$\{B \in \mathcal{S} : x \in B\}$. Because S is T_0, we have $\mathcal{S}_x \neq \mathcal{S}_y$ when $x \neq y$, and so the set $\mathcal{S}_x$ uniquely determines the point x. It is therefore convenient to define representations on topological spaces under the assumption that the space is T_0. In the specific cases below, we often have much more structure, which we use to simplify the representations.

We now develop these definitions more formally.

Definition IV.6 (Computable topological space)**.** Let S be a second-countable T_0 topological space with a countable subbasis $\mathcal{S}$. Let $s : \omega \to \mathcal{S}$ be an enumeration of $\mathcal{S}$. We say that S is a *computable topological space (with respect to s)* when the set

$$\big\{\langle m, n\rangle : s(m) = s(n)\big\} \tag{61}$$

is a c.e. subset of ω, where $\langle \cdot, \cdot \rangle$ is a standard pairing function.

This definition of a computable topological space is derived from Weihrauch [Wei00b, Def. 3.2.1]. (See also, e.g., Grubba, Schröder, and Weihrauch [GSW07, Def. 3.1].)

It is often possible to pick a subbasis $\mathcal{S}$ (and enumeration s) for which the elemental "observations" that one can computably observe are those of the form $x \in B$, where $B \in \mathcal{S}$. Then the set $\mathcal{S}_x = \{B \in \mathcal{S} : x \in B\}$ is computably enumerable (with respect to s) when one can eventually list every basic open set that contains the point x; we will call such a point x *computable*. This is one motivation for the definition of computable point in a T_0 space below.

Note that in a T_1 space, two computable points are computably distinguishable, but in a T_0 space, computable points will be, in general, distinguishable only in a computably enumerable fashion. However, this is essentially the best that is possible, if the open sets are those that we can "observe". (For more details on this approach to considering datatypes as topological spaces, in which basic open sets correspond to "observations", see Battenfeld, Schröder, and Simpson [BSS07, §2].) Note that the choice of topology and subbasis are essential; for example, we can recover both computable reals and c.e. reals as instances of "computable point" for appropriate computable topological spaces, as we describe in Section 2.1.

Definition IV.7 (Names and computable points)**.** Let $(S, \mathcal{S})$ be a computable topological space with respect to an enumeration s. Let $x \in S$. The set

$$\{n : s(n) \in \mathcal{S}_x\} \tag{62}$$

is called the *s-name* (or simply, *name*) of x. We say that x is computable when its s-name is c.e.

Note that this use of the term "name" is similar to the notion of a "complete name" (see [Wei00b, Lem. 3.2.3]), but differs somewhat from TTE usage (see [Wei00b, Def. 3.2.2]).

Definition IV.8 (Computable functions)**.** Let $(S, \mathcal{S})$ and $(T, \mathcal{T})$ be computable topological spaces (with respect to enumerations s and t, respectively). We say that a function $f : S \to T$ is *computable (with respect to s and t)* when there is a computable functional $g : \omega^\omega \to \omega^\omega$ such that for all $x \in \mathrm{dom}(f)$ and enumerations $N = \{n_i\}_{i\in\omega}$ of an s-name of x, we have that $g(N)$ is an enumeration of a t-name of $f(x)$.

(See [Wei00b, Def. 3.1.3] for more details.) Note that an implication of this definition is that computable functions are continuous.

Recall that a functional $g : \omega^\omega \to \omega^\omega$ is computable if there is a monotone computable function $h : \omega^{<\omega} \to \omega^{<\omega}$ mapping finite prefixes (i.e., finite sequences of integers) to finite prefixes, such that given increasing prefixes of an input N in the domain of g, the output of h will eventually include every finite prefix of $g(N)$. (See [Wei00b, Def. 2.1.11] for more details.) Informally, h can be used to read in an enumeration of an s-name of a point x and output an enumeration of a t-name of the point $f(x)$.

Let $(S, \mathcal{S})$ and $(T, \mathcal{T})$ be computable topological spaces. In many situations where we are interested in establishing the computability of some function $f : S \to T$, we may refer to the function implicitly via pairs of points $x \in S$ and $y \in T$ related by $y = f(x)$. In this case, we will say that y (under the topology $\mathcal{T}$) is *computable relative to x* (under the topology $\mathcal{S}$) when $f : S \to T$ is a computable function. We will often elide one or both topologies when they are clear from context.

2.1. Representations of Reals. We will use both the standard topology and right order topology on the real line $\mathbb{R}$. The reals under the standard topology are a computable topological space using the basis $\mathcal{I}_\mathbb{Q}$ with respect to a straightforward effective enumeration. The reals under the *right order topology* are a computable topological space using the basis

$$\mathcal{R}_< := \big\{(c, \infty) \,:\, c \in \mathbb{Q}\big\}, \tag{63}$$

under a standard enumeration.

Recall that, for $k \geq 1$, the set $\mathcal{I}_\mathbb{Q}^k$ is a basis for the (product of the) standard topology on $\mathbb{R}^k$ that is closed under intersection and makes $(\mathbb{R}^k, \mathcal{I}_\mathbb{Q}^k)$ a computable topological space (under a straightforward enumeration of $\mathcal{I}_\mathbb{Q}^k$). Likewise, an effective enumeration of cylinders $\sigma \times \mathbb{R}^\omega$, for $\sigma \in \bigcup_{k\geq 1} \mathcal{I}_\mathbb{Q}^k$, makes $\mathbb{R}^\omega$ a computable topological space.

Replacing $\mathcal{I}_{\mathbb{Q}}$ with $\mathcal{R}_<$ and "standard" with "right order" above gives a characterization of computable vectors and sequences of reals under the right order topology.

We can use the right order topology to define a representation for open sets. Let $(S, \mathcal{S})$ be a computable topological space, with respect to an enumeration s. Then an open set $B \subseteq S$ is *c.e. open* when the indicator function $\mathbf{1}_B$ is computable with respect to $\mathcal{S}$ and $\mathcal{R}_<$. The c.e. open sets can be shown to be the computable points in the space of open sets under the Scott topology. Note that for the computable topological space ω (under the discrete topology and the identity enumeration) the c.e. open sets are precisely the c.e. sets of integers.

2.2. Representations of Continuous Real Functions. We now consider computable representations for continuous functions on the reals.

Let $(S, \mathcal{S})$ and $(T, \mathcal{T})$ each be either of $(\mathbb{R}, \mathcal{I}_{\mathbb{Q}})$ or $(\mathbb{R}, \mathcal{R}_<)$, and let s and t be the associated enumerations. For $k \geq 1$, the compact-open topology on the space of continuous functions from S^k to T has a subbasis composed of sets of the form

$$\big\{f \,:\, f\big(\overline{A}\big) \subseteq B\big\}, \tag{64}$$

where A and B are elements in the *bases* $\mathcal{S}^k$ and $\mathcal{T}$, respectively. An effective enumeration of this subbasis can be constructed in a straightforward fashion from s and t.

In particular, let $k \geq 1$ and let s^k be an effective enumeration of k-tuples of basis elements derived from s. Then a continuous function $f : (\mathbb{R}^k, \mathcal{S}^k) \to (\mathbb{R}, \mathcal{T})$ is computable (under the compact-open topology) when

$$\big\{\langle m, n\rangle \,:\, f\big(\overline{s^k(m)}\big) \subseteq t(n)\big\} \tag{65}$$

is a c.e. set. The set (65) is the name of f.

A continuous function is computable in this sense if and only if it is computable according to Definition IV.8. (See [Wei00b, Ch. 6] and [Wei00b, Thm. 3.2.14]). Note that when $\mathcal{S} = \mathcal{T} = \mathcal{I}_{\mathbb{Q}}$, this recovers the standard definition of a computable real function. When $\mathcal{S} = \mathcal{I}_{\mathbb{Q}}$ and $\mathcal{T} = \mathcal{R}_<$, this recovers the standard definition of a lower-semicomputable real function [WZ00].

2.3. Representations of Borel Probability Measures. The following representations for probability measures on computable topological spaces are devised from more general TTE representations in Schröder [Sch07] and Bosserhoff [Bos08], and agree with Weihrauch

[Wei99] in the case of the unit interval. In particular, the representation for $\mathcal{M}_1(S)$ below is admissible with respect to the weak topology, hence computably equivalent (see Weihrauch [Wei00b, Chap. 3]) to the canonical TTE representation for Borel measures given in Schröder [Sch07].

Schröder [Sch07] has also shown the equivalence of this representation for probability measures (as a computable space under the weak topology) with *probabilistic processes*. A probabilistic process (see Schröder and Simpson [SS06]) formalizes a notion of a program that uses randomness to sample points in terms of their names of the form (62).

For a second-countable T_0 topological space S with subbasis $\mathcal{S}$, let $\mathcal{M}_1(S)$ denote the set of Borel probability measures on S (i.e., the probability measures on the σ-algebra generated by $\mathcal{S}$). Such measures are determined by the measure they assign to finite intersections of elements of $\mathcal{S}$. Note that $\mathcal{M}_1(S)$ is itself a second-countable T_0 space.

Now let $(S, \mathcal{S})$ be a computable topological space with respect to the enumeration s. We will describe a subbasis for $\mathcal{M}_1(S)$ that makes it a computable topological space. Let $\mathcal{L}_{\mathcal{S}}$ denote the lattice generated by $\mathcal{S}$ (i.e., the closure of $\mathcal{S}$ under finite union and intersection), and let $s^{\mathcal{L}}$ be an effective enumeration derived from s. Then, the class of sets

$$\{\gamma \in \mathcal{M}_1(S) \,:\, \gamma\sigma > q\}, \tag{66}$$

where $\sigma \in \mathcal{L}_{\mathcal{S}}$ and $q \in \mathbb{Q}$, is a subbasis for the weak topology on $\mathcal{M}_1(S)$. An effective enumeration of this subbasis can be constructed in a straightforward fashion from the enumeration of $\mathcal{S}$ and an effective enumeration $\{q_n\}_{n\in\omega}$ of the rationals, making $\mathcal{M}_1(S)$ a computable topological space. In particular, the name of a measure $\eta \in \mathcal{M}_1(S)$ is the set $\{\langle m, n\rangle \,:\, \eta\big(s^{\mathcal{L}}(m)\big) > q_n\}$.

Corollary IV.9 (Computable distribution)**.** *A Borel probability measure $\eta \in \mathcal{M}_1(S)$ is* computable (under the weak topology) *if and only if ηB is a c.e. real, uniformly in the $s^{\mathcal{L}}$-index of $B \in \mathcal{L}_{\mathcal{S}}$.*

Note that, for computable topological spaces $(S, \mathcal{S})$ and $(T, \mathcal{T})$ with enumerations s and t, a measure $\eta \in \mathcal{M}_1(T)$ is computable relative to a point $x \in S$ when ηB is a c.e. real relative to x, uniformly in the $t^{\mathcal{L}}$-index of $B \in \mathcal{L}_{\mathcal{T}}$. Corollary IV.9 implies that the measure of a c.e. open set (i.e., the c.e. union of basic open sets) is a c.e. real (uniformly in the enumeration of the terms in the union), and that the measure of a co-c.e. closed set (i.e., the complement of a c.e. open set) is a co-c.e. real (similarly uniformly); see, e.g., [BP03, §3.3] for details. Note that on a discrete space, where singletons are both c.e. open and

co-c.e. closed, the measure of each singleton is a computable real. But for a general space, it is too strong to require that even basic open sets have computable measure (see Weihrauch [Wei99] for a discussion; moreover, such a requirement is stronger than necessary to ensure that a probabilistic Turing machine can produce exact samples to arbitrary accuracy).

We will be interested in computable measures in $\mathcal{M}_1(S)$, where S is either $\mathbb{R}^\omega$, $[0,1]^k$, or $\mathcal{M}_1(\mathbb{R})$. In order to apply Corollary IV.9 to characterize concrete notions of computability for $\mathcal{M}_1(S)$, we will now describe choices of topologies on these three spaces.

Measures on Real Vectors and Sequences under the Standard Topology. Using Corollary IV.9, we can characterize the class of computable distributions on real sequences using the computable topological spaces characterized above in Section 2.1. Let $\vec{x} = \{x_i\}_{i\geq 1}$ be a sequence of real-valued random variables (e.g., the exchangeable sequence X, or the derived random variables $\{V_\tau\}_{\tau\in\mathcal{I}_\mathbb{Q}}$ under the canonical enumeration of $\mathcal{I}_\mathbb{Q}$), and let η be the joint distribution of $\vec{x}$. Then η is computable if and only if $\eta(\sigma\times\mathbb{R}^\omega) = \mathbf{P}\big\{x\in\sigma\times\mathbb{R}^\omega\big\}$ is a c.e. real, uniformly in $k\geq 1$ and $\sigma\in\mathcal{L}_{\mathbb{Q}^k}$. The following simpler characterization was given by Müller [Mül99, Thm. 3.7].

Lemma IV.10 (Computable distribution under the standard topology)**.** *Let $\vec{x} = \{x_i\}_{i\geq 1}$ be a sequence of real-valued random variables with joint distribution η. Then η is computable if and only if*

$$\eta(\tau\times\mathbb{R}^\omega) = \mathbf{P}\big(\textstyle\bigcap_{i=1}^k\{x_i\in\tau(i)\}\big) \tag{67}$$

is a c.e. real, uniformly in $k\geq 1$ and $\tau\in\mathcal{I}_\mathbb{Q}^k$.

Therefore knowing the measure of the sets in $\bigcup_k\mathcal{I}_\mathbb{Q}^k\subsetneq\bigcup_k\mathcal{L}_{\mathbb{Q}^k}$ is sufficient. Note that the right-hand side of (67) is precisely the form of the left-hand side of the expression in Corollary IV.5. Note also that one obtains a characterization of the computability of a finite-dimensional vector by embedding it as an initial segment of a sequence.

Measures on Real Vectors and Sequences under the Right Order Topology. Borel measures on $\mathbb{R}$ under the right order topology play an important role when representing measures on measures, as Corollary IV.9 portends.

Corollary IV.11 (Computable distribution under the right order topology)**.** *Let $\vec{x} = \{x_i\}_{\imath\geq 1}$ be a sequence of real-valued random variables with joint distribution η. Then η is computable under the (product of*

the) right order topology if and only if

$$\eta\big(\bigcup_{i=1}^{m}\big((c_{i1},\infty)\times\cdots\times(c_{ik},\infty)\times\mathbb{R}^{\omega}\big)\big) = \mathbf{P}\big(\bigcup_{i=1}^{m}\bigcap_{j=1}^{k}\{x_j > c_{ij}\}\big) \quad (68)$$

is a c.e. real, uniformly in $k, m \geq 1$ and $C = (c_{ij}) \in \mathbb{Q}^{m\times k}$.

Again, one obtains a characterization of the computability of a finite-dimensional vector by embedding it as an initial segment of a sequence. Note also that if a distribution on $\mathbb{R}^k$ is computable under the standard topology, then it is clearly computable under the right order topology. The above characterization is used in the next section as well as in Proposition IV.17, where we must compute an integral with respect to a topology that is weaker than the standard topology.

Measures on Borel Measures. The de Finetti measure μ is the distribution of the directing random measure ν, an $\mathcal{M}_1(\mathbb{R})$-valued random variable. Recall the definition $V_{\boldsymbol{\beta}} := \nu\boldsymbol{\beta}$, for $\boldsymbol{\beta} \in \mathcal{B}_{\mathbb{R}}$. From Corollary IV.9, it follows that μ is computable under the weak topology if and only if

$$\mu\big(\bigcup_{i=1}^{m}\bigcap_{j=1}^{k}\{\gamma\in\mathcal{M}_1(\mathbb{R}) \,:\, \gamma\sigma(j) > c_{ij}\}\big) = \mathbf{P}\big(\bigcup_{i=1}^{m}\bigcap_{j=1}^{k}\{V_{\sigma(j)} > c_{ij}\}\big) \quad (69)$$

is a c.e. real, uniformly in $k, m \geq 1$ and $\sigma \in \mathcal{L}_{\mathbb{Q}}^{k}$ and $C = (c_{ij}) \in \mathbb{Q}^{m\times k}$. As an immediate consequence of (69) and Corollary IV.11, we obtain the following characterization of computable de Finetti measures.

Corollary IV.12 (Computable de Finetti measure)**.** *The de Finetti measure μ is computable relative to the joint distribution of $\{V_\tau\}_{\tau\in\mathcal{L}_{\mathbb{Q}}}$ under the right order topology, and vice versa. In particular, μ is computable if and only if the joint distribution of $\{V_\tau\}_{\tau\in\mathcal{L}_{\mathbb{Q}}}$ is computable under the right order topology.*

Integration. The following lemma is a restatement of an integration result by Schröder [Sch07, Prop. 3.6], which itself generalizes integration results on standard topologies of finite-dimensional Euclidean spaces by Müller [Mül99] and the unit interval by Weihrauch [Wei99].

Define

$$\mathbb{I} := \{A\cap[0,1] \,:\, A\in\mathcal{I}_{\mathbb{Q}}\}, \quad (70)$$

which is a basis for the standard topology on $[0,1]$, and define

$$\mathbb{I}_{<} := \{A\cap[0,1] \,:\, A\in\mathcal{R}_{<}\}, \quad (71)$$

which is a basis for the right order topology on $[0,1]$.

Lemma IV.13 (Integration of bounded lower-semicontinuous functions)**.** *Let $k \geq 1$ and let $\mathcal{S}$ be either $\mathcal{I}_{\mathbb{Q}}$ or $\mathcal{R}_<$. Let*

$$f : (\mathbb{R}^k, \mathcal{S}^k) \to ([0,1], \mathbb{I}_<) \tag{72}$$

be a continuous function and let μ be a Borel probability measure on $(\mathbb{R}^k, \mathcal{S}^k)$. Then

$$\int f \, d\mu \tag{73}$$

is a c.e. real relative to f and μ.

The following result of Müller [Mül99] is an immediate corollary.

Corollary IV.14 (Integration of bounded continuous functions)**.** *Let*

$$g : (\mathbb{R}^k, \mathcal{I}^k_{\mathbb{Q}}) \to ([0,1], \mathbb{I}) \tag{74}$$

be a continuous function and let μ be a Borel probability measure on $(\mathbb{R}^k, \mathcal{I}^k_{\mathbb{Q}})$. Then

$$\int g \, d\mu \tag{75}$$

is a computable real relative to g and μ.

3. The Computable Moment Problem

One often has access to the moments of a distribution, and wishes to recover the underlying distribution. Let $\vec{x} = (x_i)_{i \in \omega}$ be a random vector in $[0,1]^\omega$ with distribution η. Classically, the distribution of $\vec{x}$ is uniquely determined by the mixed moments of $\vec{x}$. We show that the distribution is in fact *computable* from the mixed moments.

One classical way to pass from the moments of $\vec{x}$ to its distribution is via the Lévy inversion formula, which maps the characteristic function $\phi_{\vec{x}} : \mathbb{R}^\omega \to \mathbf{C}$, given by

$$\phi_{\vec{x}}(t) := \mathbb{E}(e^{i\langle t, \vec{x}\rangle}), \tag{76}$$

to the distribution of $\vec{x}$. However, even in the finite-dimensional case, the inversion formula involves a limit for which we have no direct handle on the rate of convergence, and so the distribution it defines is not obviously computable. Instead, we use a computable version of the Weierstrass approximation theorem to compute the distribution relative to the mixed moments.

To show that η is computable relative to the mixed moments, it suffices to show that $\eta(\sigma \times [0,1]^\omega) = \mathbb{E}\big(\mathbf{1}_\sigma(x_1, \ldots, x_k)\big)$ is a c.e. real

relative to the mixed moments, uniformly in $\sigma \in \bigcup_{k\geq 1} \mathcal{I}^k_{\mathbb{Q}}$. We begin by building sequences of polynomials that converge pointwise from below to indicator functions of the form $\mathbf{1}_\sigma$ for $\sigma \in \bigcup_{k\geq 1} \mathcal{L}_{\mathbb{Q}^k}$ (see Figure 1.)

Lemma IV.15 (Polynomial approximations)**.** *Let $k \geq 1$ and $\sigma \in \mathcal{L}_{\mathbb{Q}^k}$. There is a sequence*

$$\big\{p_{n,\sigma} \,:\, n \in \omega\big\} \tag{77}$$

of rational polynomials of degree k, computable uniformly in n, k, and σ, such that, for all $\vec{x} \in [0,1]^k$, we have

$$-2 \leq p_{n,\sigma}(\vec{x}) \leq \mathbf{1}_\sigma(\vec{x}) \qquad \textit{and} \qquad \lim_{m\to\infty} p_{m,\sigma}(\vec{x}) = \mathbf{1}_\sigma(\vec{x}). \tag{78}$$

Proof. Let $k \geq 1$. For $\sigma \in \mathcal{L}_{\mathbb{Q}^k}$, and $\vec{x} \in \mathbb{R}^k$, define $d(\vec{x}, [0,1]^k \setminus \sigma)$ to be the distance from $\vec{x}$ to the nearest point in $[0,1]^k \setminus \sigma$. It is straightforward to show that $d(\vec{x}, [0,1]^k \setminus \sigma)$ is a computable real function of $\vec{x}$, uniformly in k and σ.

For $n \in \omega$, define $f_{n,\sigma} : \mathbb{R}^k \to \mathbb{R}$ by

$$f_{n,\sigma}(\vec{x}) := -\frac{1}{n+1} + \min\{1, n \cdot d(\vec{x}, [0,1]^k \setminus \sigma)\}, \tag{79}$$

and note that $-1 \leq f_{n,\sigma}(\vec{x}) \leq \mathbf{1}_\sigma(\vec{x}) - \frac{1}{n+1}$ and $\lim_{m\to\infty} f_{m,\sigma}(\vec{x}) = \mathbf{1}_\sigma(\vec{x})$. Furthermore, $f_{n,\sigma}(\vec{x})$ is a computable (hence continuous) real function of $\vec{x}$, uniformly in n, k, and σ.

By the effective Weierstrass approximation theorem (see Pour-El and Richards [PER89, p. 45]), we can find (uniformly in n, k, and σ) a polynomial $p_{n,\sigma}$ with rational coefficients that uniformly approximates $f_{n,\sigma}$ to within $1/(n+1)$ on $[0,1]^k$. These polynomials have the desired properties. □

We thank an anonymous referee for suggestions that simplified the proof of this lemma.

Using these polynomials, we can compute the distribution from the moments. (See Figure 1 for a depiction of the moment based approximation.) The other direction follows from computable integration results.

Theorem IV.16 (Computable moments)**.** *Let $\vec{x} = (x_i)_{i\in\omega}$ be a random vector in $[0,1]^\omega$ with distribution η. Then η is computable relative to the mixed moments of $\{x_i\}_{i\in\omega}$, and vice versa. In particular, η is computable if and only if the mixed moments of $\{x_i\}_{i\in\omega}$ are uniformly computable.*

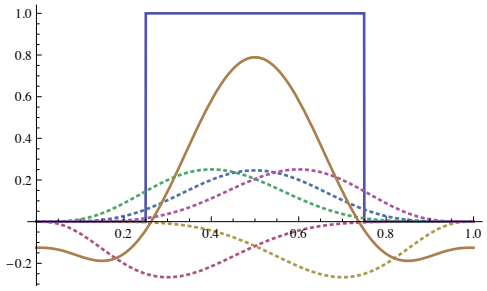

FIGURE 1. A polynomial approximation (solid curve, produced by summation of dashed polynomial curves) to the indicator function for the interval $(\frac{1}{4}, \frac{3}{4})$. Expectations of polynomials correspond to linear combinations of moments, while the expectation of the indicator function is simply the probability measure of the interval. As the polynomial approximations improve, so does our estimate of the interval's probability. We can construct a sequence of polynomial approximations (like the solid curve) that converges pointwise from below to the indicator function. This shows that the associated interval probability is a c.e. real, provided that the moments themselves are uniformly computable.

PROOF. Any monic monomial in k variables, considered as a real function, computably maps $[0,1]^k$ into $[0,1]$ (under the standard topology). Furthermore, as the restriction of η to any k coordinates is computable relative to η (uniformly in the coordinates), it follows from Corollary IV.14 that each mixed moment (the expectation of a monomial under such a restriction of η) is computable relative to η, uniformly in the index of the monomial and the coordinates.

Let $k \geq 1$ and $\sigma \in \mathcal{I}^k_{\mathbb{Q}}$. To establish the computability of η, it suffices to show that

$$\eta(\sigma \times [0,1]^\omega) = \mathbb{E}\big(\mathbf{1}_{\sigma\times[0,1]^\omega}(\vec{x})\big) = \mathbb{E}\big(\mathbf{1}_\sigma(x_1,\ldots,x_k)\big). \tag{80}$$

is a c.e. real relative to the mixed moments, uniformly in k and σ. By Lemma IV.15, there is a uniformly computable sequence of polynomials $(p_{n,\sigma})_{n\in\omega}$ that converge pointwise from below to the indicator $\mathbf{1}_\sigma$. Therefore, by the dominated convergence theorem,

$$\mathbb{E}\big(\mathbf{1}_\sigma(x_1,\ldots,x_k)\big) = \sup_n \mathbb{E}\big(p_{n,\sigma}(x_1,\ldots,x_k)\big). \tag{81}$$

The expectation $\mathbb{E}\big(p_{n,\sigma}(x_1,\ldots,x_k)\big)$ is a $\mathbb{Q}$-linear combination of mixed moments, hence a computable real relative to the mixed moments,

uniformly in n, k, and σ. Thus the supremum (81) is a c.e. real relative to the mixed moments, uniformly in k and σ. □

4. Proof of the Computable de Finetti Theorem

For the remainder of the chapter, let X be a real-valued exchangeable sequence with distribution χ, let ν be its directing random measure, and let μ be the corresponding de Finetti measure.

Classically, the joint distribution of X is uniquely determined by the de Finetti measure (see Equation 55). We now show that the joint distribution of X is in fact *computable* relative to the de Finetti measure.

Proposition IV.17. *The distribution χ is computable relative to μ.*

Proof. Let $k \geq 1$ and $\sigma \in \mathcal{I}^k_{\mathbb{Q}}$. All claims are uniform in k and σ. In order to show that χ, the distribution of X, is computable relative to μ, we must show that $\mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \sigma(i)\}\big)$ is a c.e. real relative to μ. Note that, by Corollary IV.5,

$$\mathbf{P}\big(\textstyle\bigcap_{i=1}^k \{X_i \in \sigma(i)\}\big) = \mathbb{E}\big(\prod_{i=1}^k V_{\sigma(i)}\big). \tag{82}$$

Let η be the joint distribution of $(V_{\sigma(i)})_{i \leq k}$ and let $f : [0,1]^k \to [0,1]$ be defined by

$$f(x_1, \ldots, x_k) := \textstyle\prod_{i=1}^k x_i. \tag{83}$$

To complete the proof, we now show that

$$\int f \, d\eta = \mathbb{E}\big(\textstyle\prod_{i=1}^k V_{\sigma(i)}\big) \tag{84}$$

is a c.e. real relative to μ. Note that η is computable under the right order topology relative to μ. Furthermore, f is order-preserving (in each dimension) and lower-semicontinuous, i.e., is a continuous (and obviously computable) function from $([0,1]^k, \mathbb{I}^k_<)$ to $([0,1], \mathbb{I}_<)$. Therefore, by Lemma IV.13, we have that $\int f \, d\eta$ is a c.e. real relative to μ. □

We will first prove the main theorem under the additional hypothesis that the directing random measure is almost surely continuous. We then sketch a randomized argument that succeeds with probability one. Finally, we present the proof of the main result, which can be seen as a derandomization.

4.1. Almost Surely Continuous Directing Random Measures. For $k \geq 1$ and $\boldsymbol{\psi} \in \mathcal{L}^k_{\mathbb{R}}$, we say that $\boldsymbol{\psi}$ is a *ν-continuity set* when, for $i \leq k$, we have $\nu(\partial\boldsymbol{\psi}(i)) = 0$ a.s., where $\partial\boldsymbol{\psi}(i)$ denotes the boundary of $\boldsymbol{\psi}(i)$.

Lemma IV.18. *Relative to χ, the mixed moments of $\{V_\tau\}_{\tau\in\mathcal{L}_\mathbb{Q}}$ are uniformly c.e. reals and the mixed moments of $\{V_{\overline{\tau}}\}_{\tau\in\mathcal{L}_\mathbb{Q}}$ are uniformly co-c.e. reals; in particular, if $\sigma \in \mathcal{L}^k_\mathbb{Q}$ (for $k \geq 1$) is a ν-continuity set, then the mixed moment $\mathbb{E}\big(\prod_{i=1}^k V_{\sigma(i)}\big)$ is a computable real, uniformly in k and σ.*

Proof. Let $k \geq 1$ and $\sigma \in \mathcal{L}^k_\mathbb{Q}$. All claims are uniform in k and σ. By Corollary IV.5,

$$\mathbb{E}\big(\textstyle\prod_{i=1}^k V_{\sigma(i)}\big) = \mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \sigma(i)\}\big), \tag{85}$$

which is a c.e. real relative to χ. The set $\overline{\sigma}$ is a co-c.e. closed set in $\mathbb{R}^k$ because we can computably enumerate all $\tau \in \mathcal{L}^k_\mathbb{Q}$ contained in the complement of σ. Therefore,

$$\mathbb{E}\big(\textstyle\prod_{i=1}^k V_{\overline{\sigma(i)}}\big) = \mathbf{P}\big(\bigcap_{i=1}^k \{X_i \in \overline{\sigma(i)}\}\big) \tag{86}$$

is the measure of a co-c.e. closed set, hence a co-c.e. real relative to χ. When σ is a ν-continuity set,

$$\mathbb{E}\big(\textstyle\prod_{i=1}^k V_{\sigma(i)}\big) = \mathbb{E}\big(\prod_{i=1}^k V_{\overline{\sigma(i)}}\big), \tag{87}$$

and so the expectation is a computable real relative to χ. □

Proposition IV.19 (Almost surely continuous directing random measure)**.** *Assume that ν is almost surely continuous. Then μ is computable relative to χ.*

Proof. Let $k \geq 1$ and $\sigma \in \mathcal{L}^k_\mathbb{Q}$. The almost sure continuity of ν implies that σ is an ν-continuity set. Therefore, by Lemma IV.18, the moment $\mathbb{E}\big(\prod_{i=1}^k V_{\sigma(i)}\big)$ is a computable real relative to χ, uniformly in k and σ. The computable moment theorem (Theorem IV.16) then implies that the joint distribution of the variables $\{V_\tau\}_{\tau\in\mathcal{L}_\mathbb{Q}}$ is computable under the standard topology relative to χ, and so their joint distribution is also computable under the (weaker) right order topology relative to χ. By Corollary IV.12, this implies that μ is computable relative to χ. □

4.2. "Randomized" Proof Sketch. In general, the joint distribution of $\{V_\sigma\}_{\sigma\in\mathcal{L}_\mathbb{Q}}$ is not computable under the standard topology because the directing random measure ν may, with nonzero probability, have a point mass on a rational. In this case, the mixed moments of $\{V_\tau\}_{\tau\in\mathcal{L}_\mathbb{Q}}$ are c.e., but not co-c.e., reals relative to χ. As a result, the

computable moment theorem (Theorem IV.16) is inapplicable. For arbitrary directing random measures, we give a proof of the computable de Finetti theorem that works regardless of the location of point masses.

Consider the following sketch of a "randomized algorithm": We independently sample a countably infinite sequence of real numbers $\mathbf{A}$ from a computable, absolutely continuous distribution that has support everywhere on the real line (e.g., a Gaussian or Cauchy). Let $\mathcal{L}_{\mathbf{A}}$ denote the lattice generated by open intervals with endpoints in $\mathbf{A}$. Note that, with probability one, $\mathbf{A}$ will be dense in $\mathbb{R}$ and every $\boldsymbol{\psi} \in \mathcal{L}_{\mathbf{A}}$ will be a ν-continuity set. If the algorithm proceeds analogously to the case where ν is almost surely continuous, using $\mathcal{L}_{\mathbf{A}}$ as our basis, rather than $\mathcal{L}_{\mathbb{Q}}$, then it will compute the de Finetti measure with probability one.

Let A be a dense sequence of reals such that $\nu(A) = 0$ a.s. Consider the variables $V_{\boldsymbol{\zeta}}$ defined in terms of elements $\boldsymbol{\zeta}$ of the new basis $\mathcal{L}_A$ (defined analogously to $\mathcal{L}_{\mathbf{A}}$). We begin by proving an extension of Lemma IV.18: The mixed moments of the set of variables $\{V_\zeta\}_{\zeta \in \mathcal{L}_A}$ are computable relative to A and χ.

Lemma IV.20. *Let $k \geq 1$ and $\psi \in \mathcal{L}_A^k$. The mixed moment $\mathbb{E}\big(\prod_{i=1}^k V_{\psi(i)}\big)$ is a computable real relative to A and χ, uniformly in k and ψ.*

Proof. Let $k \geq 1$ and $\psi \in \mathcal{L}_A^k$. All claims are uniform in k and ψ. We first show that, relative to A and χ, the mixed moments of $\{V_\zeta\}_{\zeta \in \mathcal{L}_A}$ are uniformly c.e. reals. We can compute (relative to A) a sequence

$$\sigma_1, \sigma_2, \ldots \in \mathcal{L}_{\mathbb{Q}}^k \tag{88}$$

such that for each $n \geq 1$,

$$\sigma_n \subseteq \sigma_{n+1} \qquad \text{and} \qquad \bigcup_m \sigma_m = \psi. \tag{89}$$

Note that if $\zeta, \varphi \in \mathcal{L}_{\mathbb{Q}}$ satisfy $\zeta \subseteq \varphi$, then $V_\zeta \leq V_\varphi$ (a.s.), and so, by the continuity of measures (and of multiplication), $\prod_{i=1}^k V_{\sigma_n(i)}$ converges from below to $\prod_{i=1}^k V_{\psi(i)}$ with probability one. Therefore, the dominated convergence theorem gives us

$$\mathbb{E}\big(\textstyle\prod_{i=1}^k V_{\psi(i)}\big) = \sup_n \mathbb{E}\big(\prod_{i=1}^k V_{\sigma_n(i)}\big). \tag{90}$$

Using Corollary IV.5, we see that the expectation $\mathbb{E}\big(\prod_{i=1}^k V_{\sigma_n(i)}\big)$ is a c.e. real relative to A and χ, uniformly in n, and so the supremum (90) is a c.e. real relative to A and χ.

Similarly, the mixed moments of $\{V_{\overline{\zeta}}\}_{\zeta \in \mathcal{L}_A}$ are uniformly co-c.e. reals relative to A and χ, as can be seen via a sequence of nested unions of rational intervals whose intersection is $\overline{\psi}$. Thus, because ψ is a ν-continuity set, the mixed moment $\mathbb{E}\big(\prod_{i=1}^k V_{\psi(i)}\big)$ is a computable real relative to A and χ. □

Lemma IV.21. *The de Finetti measure μ is computable relative to A and χ.*

PROOF. It follows immediately from Lemma IV.20 and Theorem IV.16 that the joint distribution of $\{V_\psi\}_{\psi\in\mathcal{L}_A}$ is computable relative to A and χ. This joint distribution classically determines the de Finetti measure. Moreover, as we now show, we can compute (relative to A and χ) the desired representation with respect to the (original) rational basis. In particular, we prove that the joint distribution of $\{V_\tau\}_{\tau\in\mathcal{L}_\mathbb{Q}}$ is computable under the right order topology relative to A and χ.

Let $m, k \geq 1$, let $\tau \in \mathcal{L}^k_\mathbb{Q}$, and let $C = (c_{ij}) \in \mathbb{Q}^{m\times k}$. We will express τ as a union of elements of $\mathcal{L}^k_A$. Note that τ is an c.e. open set (relative to A) with respect to the basis $\mathcal{L}^k_A$. In particular, we can computably enumerate (relative to A, and uniformly in k and τ) a sequence $\sigma_1, \sigma_2, \ldots \in \mathcal{L}^k_A$ such that $\cup_n \sigma_n = \tau$ and $\sigma_n \subseteq \sigma_{n+1}$. Note that $V_{\tau(j)} \geq V_{\sigma_n(j)}$ (a.s.) for all $n \geq 1$ and $j \leq k$. By the continuity of measures (and of union and intersection),

$$\mathbf{P}\big(\textstyle\bigcup_{i=1}^m \bigcap_{j=1}^k \{V_{\tau(j)} > c_{ij}\}\big) = \sup_n \mathbf{P}\big(\textstyle\bigcup_{i=1}^m \bigcap_{j=1}^k \{V_{\sigma_n(j)} > c_{ij}\}\big). \tag{91}$$

The probability $\mathbf{P}\big(\bigcup_{i=1}^m \bigcap_{j=1}^k \{V_{\sigma_n(j)} > c_{ij}\}\big)$ is a c.e. real relative to A and χ, uniformly in n, m, k, τ, and C, and so the supremum (91) is a c.e. real relative to A and χ, uniformly in m, k, τ, and C. □

Let Φ denote the map taking (A, χ) to μ, as described in Lemma IV.21.

Recall that $\mathbf{A}$ is a random dense sequence with a computable distribution, as defined above, and let $\hat{\mu} = \Phi(\mathbf{A}, \chi)$. Then $\hat{\mu}$ is a random variable, and moreover, $\hat{\mu} = \mu$ almost surely. However, while $\mathbf{A}$ is almost surely noncomputable, the distribution of $\mathbf{A}$ is computable, and so the distribution of $\hat{\mu}$ is computable relative to χ. Expectations with respect to the distribution of $\hat{\mu}$ can then be used to (deterministically) compute μ relative to χ.

A proof along these lines could be made precise by making

$$\mathcal{M}_1(\mathcal{M}_1(\mathcal{M}_1(\mathbb{R}))) \tag{92}$$

into a computable topological space. Instead, in Section 4.3, we complete the proof by explicitly computing μ relative to χ in terms of the standard rational basis. This construction can be seen as a "derandomization" of the above algorithm.

Alternatively, the above sketch could be interpreted as a degenerate *probabilistic process* (see Schröder and Simpson [SS06]) that samples a name of the de Finetti measure with probability one. Schröder

[Sch07] shows that representations in terms of probabilistic processes are computably reducible to representations of computable distributions.

The structure of the derandomized argument occurs in other proofs in computable analysis and probability theory. Weihrauch [Wei99, Thm. 3.6] proves a computable integration result via an argument that could likewise be seen as a derandomization of an algorithm that densely subdivides the unit interval at random locations to find continuity sets. Bosserhoff [Bos08, Lem. 2.15] uses a similar argument to compute a basis for a computable metric space, for which every basis element is a continuity set; this suggests an alternative approach to completing our proof. Müller [Mül99, Thm. 3.7] uses a similar construction to find open hypercubes such that for any $\epsilon > 0$, the probability on their boundaries is less than ϵ. These arguments also resemble the proof of the classical Portmanteau theorem [Kal02, Thm. 4.25], in which an uncountable family of sets with disjoint boundaries is defined, almost all of which are continuity sets.

4.3. "Derandomized" Construction. Let $m, k \geq 1$ and $C = (c_{ij}) \in \mathbb{Q}^{m\times k}$. By an abuse of notation, we define

$$\mathbf{1}_C : [0,1]^k \to [0,1] \tag{93}$$

to be the indicator function for the set

$$\textstyle\bigcup_{i=1}^{m}(c_{i1},1] \times \cdots \times (c_{ik},1]. \tag{94}$$

For $n \in \omega$, we denote by $p_{n,C}$ the polynomial $p_{n,\sigma}$ (as defined in Lemma IV.15), where

$$\sigma := \textstyle\bigcup_{i=1}^{m}(c_{i1},2) \times \cdots \times (c_{ik},2) \in \mathcal{L}_{\mathbb{Q}^k}. \tag{95}$$

Here, we have arbitrarily chosen $2 > 1$ so that the sequence of polynomials $\{p_{n,C}\}_{n\in\omega}$ converges pointwise from below to $\mathbf{1}_C$ on $[0,1]^k$.

Let $\vec{x} = (x_1, \ldots, x_k)$ and $\vec{y} = (y_1, \ldots, y_k)$. We can write

$$p_{n,C}(\vec{x}) = p^+_{n,C}(\vec{x}) - p^-_{n,C}(\vec{x}), \tag{96}$$

where $p^+_{n,C}$ and $p^-_{n,C}$ are polynomials with positive coefficients. Define the $2k$-variable polynomial

$$q_{n,C}(\vec{x},\vec{y}) := p^+_{n,C}(\vec{x}) - p^-_{n,C}(\vec{y}). \tag{97}$$

We denote

$$q_{n,C}(V_{\boldsymbol{\varphi}(1)}, \ldots, V_{\boldsymbol{\varphi}(k)}, V_{\boldsymbol{\zeta}(1)}, \ldots, V_{\boldsymbol{\zeta}(k)}) \tag{98}$$

by $q_{n,C}(V_{\boldsymbol{\varphi}}, V_{\boldsymbol{\zeta}})$, and similarly with $p_{n,C}$.

Proposition IV.22. *Let $n \in \omega$, let $k, m \geq 1$, let $\sigma \in \mathcal{L}^k_{\mathbb{Q}}$, and let $C \in \mathbb{Q}^{m\times k}$. Then $\mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}})$ is a c.e. real relative to χ, uniformly in n, k, m, σ, and C.*

Proof. By Lemma IV.18, relative to χ, and uniformly in n, k, m, σ, and C, each monomial of $p^+_{n,C}(V_\sigma)$ has a c.e. real expectation, and each monomial of $p^-_{n,C}(V_{\overline{\sigma}})$ has a co-c.e. real expectation, and so by the linearity of expectation $\mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}})$ is a c.e. real. □

In the final proof we use the following dense partial order on products of $\mathcal{L}_{\mathbb{R}}$.

Definition IV.23. Let $k \geq 1$. We call $\boldsymbol{\psi} \in \mathcal{L}^k_{\mathbb{R}}$ a *refinement* of $\boldsymbol{\varphi} \in \mathcal{L}^k_{\mathbb{R}}$, and write $\boldsymbol{\psi} \lhd \boldsymbol{\varphi}$, when

$$\overline{\boldsymbol{\psi}(i)} \subseteq \boldsymbol{\varphi}(i) \tag{99}$$

for all $i \leq k$.

We are now ready to prove the main theorem.

Proof of Computable de Finetti Theorem IV.3. The distribution χ (of the exchangeable sequence X) is computable relative to the de Finetti measure μ by Proposition IV.17. We now give a proof of the other direction, showing that the joint distribution of $\{V_\sigma\}_{\sigma\in\mathcal{L}_{\mathbb{Q}}}$ is computable under the right order topology relative to χ, which by Corollary IV.12 will complete the proof.

Let $k, m \geq 1$, let $\pi \in \mathcal{L}^k_{\mathbb{Q}}$, and let $C = (c_{ij}) \in \mathbb{Q}^{m\times k}$. For $\boldsymbol{\zeta} \in \mathcal{L}^k_{\mathbb{R}}$, let $V_{\boldsymbol{\zeta}}$ denote the k-tuple $(V_{\boldsymbol{\zeta}(1)}, \ldots, V_{\boldsymbol{\zeta}(k)})$ and similarly for $V_{\overline{\boldsymbol{\zeta}}}$. Take $\mathbf{1}_C$ to be defined as above in (93) and (94). It suffices to show that

$$\mathbf{P}\left(\bigcup_{i=1}^m \bigcap_{j=1}^k \{V_{\pi(j)} > c_{ij}\}\right) = \mathbb{E}\mathbf{1}_C(V_\pi) \tag{100}$$

is a c.e. real relative to χ, uniformly in k, m, π, and C. We do this by a series of reductions, which results in a supremum over quantities of the form $\mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}})$ for $\sigma \in \mathcal{L}^k_{\mathbb{Q}}$.

By the density of the reals and the continuity of measures, we have that

$$V_\pi = \sup_{\boldsymbol{\psi}\lhd\pi} V_{\boldsymbol{\psi}} \quad \text{a.s.}, \tag{101}$$

where $\boldsymbol{\psi}$ ranges over $\mathcal{L}^k_{\mathbb{R}}$. It follows that

$$\mathbf{1}_C(V_\pi) = \sup_{\boldsymbol{\psi}\lhd\pi} \mathbf{1}_C(V_{\boldsymbol{\psi}}) \quad \text{a.s.}, \tag{102}$$

because $\mathbf{1}_C$ is lower-semicontinuous and order-preserving (in each dimension), as (94) is an open set in the right order topology on $[0,1]^k$. Therefore, by the dominated convergence theorem, we have that

$$\mathbb{E}\mathbf{1}_C(V_\pi) = \sup_{\boldsymbol{\psi} \lhd \pi} \mathbb{E}\mathbf{1}_C(V_{\boldsymbol{\psi}}). \tag{103}$$

Recall that the polynomials $\{p_{n,C}\}_{n\in\omega}$ converge pointwise from below to $\mathbf{1}_C$ in $[0,1]^k$. Therefore, by the dominated convergence theorem,

$$\mathbb{E}\mathbf{1}_C(V_{\boldsymbol{\psi}}) = \sup_n \mathbb{E}p_{n,C}(V_{\boldsymbol{\psi}}). \tag{104}$$

As $V_{\overline{\boldsymbol{\psi}(i)}} \geq V_{\boldsymbol{\psi}(i)}$ a.s. for $i \leq k$, we have that

$$\mathbb{E}p_{n,C}(V_{\boldsymbol{\psi}}) = \mathbb{E}p^+_{n,C}(V_{\boldsymbol{\psi}}) - \mathbb{E}p^-_{n,C}(V_{\boldsymbol{\psi}}) \tag{105}$$

$$\geq \mathbb{E}p^+_{n,C}(V_{\boldsymbol{\psi}}) - \mathbb{E}p^-_{n,C}(V_{\overline{\boldsymbol{\psi}}}). \tag{106}$$

Note that if $\boldsymbol{\psi}$ is a ν-continuity set, then $V_{\overline{\boldsymbol{\psi}(i)}} = V_{\boldsymbol{\psi}(i)}$ a.s., and so

$$\mathbb{E}p_{n,C}(V_{\boldsymbol{\psi}}) = \mathbb{E}p^+_{n,C}(V_{\boldsymbol{\psi}}) - \mathbb{E}p^-_{n,C}(V_{\overline{\boldsymbol{\psi}}}). \tag{107}$$

Again, dominated convergence theorem gives us

$$\mathbb{E}\left(\prod_{i=1}^k V_{\boldsymbol{\psi}(i)}\right) = \sup_{\sigma \lhd \boldsymbol{\psi}} \mathbb{E}\left(\prod_{i=1}^k V_{\sigma(i)}\right) \quad \text{and} \tag{108}$$

$$\mathbb{E}\left(\prod_{i=1}^k V_{\overline{\boldsymbol{\psi}(i)}}\right) = \inf_{\tau \rhd \boldsymbol{\psi}} \mathbb{E}\left(\prod_{i=1}^k V_{\overline{\tau(i)}}\right), \tag{109}$$

where σ and τ range over $\mathcal{L}^k_{\mathbb{Q}}$. Therefore, by the linearity of expectation,

$$\mathbb{E}p^+_{n,C}(V_{\boldsymbol{\psi}}) = \sup_{\sigma \lhd \boldsymbol{\psi}} \mathbb{E}p^+_{n,C}(V_\sigma) \quad \text{and} \tag{110}$$

$$\mathbb{E}p^-_{n,C}(V_{\overline{\boldsymbol{\psi}}}) = \inf_{\tau \rhd \boldsymbol{\psi}} \mathbb{E}p^-_{n,C}(V_{\overline{\tau}}), \tag{111}$$

and so, if $\boldsymbol{\psi}$ is a ν-continuity set, we have that

$$\mathbb{E}p_{n,C}(V_{\boldsymbol{\psi}}) = \sup_{\sigma \lhd \boldsymbol{\psi}} \mathbb{E}p^+_{n,C}(V_\sigma) - \inf_{\tau \rhd \boldsymbol{\psi}} \mathbb{E}p^-_{n,C}(V_{\overline{\tau}}) \tag{112}$$

$$= \sup_{\sigma \lhd \boldsymbol{\psi} \lhd \tau} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\tau}}). \tag{113}$$

Because ν has at most countably many point masses, those $\boldsymbol{\psi} \in \mathcal{I}^k_{\mathbb{R}}$ that are ν-continuity sets are dense in $\mathcal{I}^k_{\mathbb{Q}}$. On the other hand, for those $\boldsymbol{\psi}$ that are not ν-continuity sets, (113) is a lower bound, as can be shown from (106). Therefore,

$$\sup_{\boldsymbol{\psi} \lhd \pi} \mathbb{E}p_{n,C}(V_{\boldsymbol{\psi}}) = \sup_{\boldsymbol{\psi} \lhd \pi} \sup_{\sigma \lhd \boldsymbol{\psi} \lhd \tau} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\tau}}). \tag{114}$$

Note that $\{(\sigma,\tau) : (\exists \boldsymbol{\psi} \lhd \pi)\ \sigma \lhd \boldsymbol{\psi} \lhd \tau\} = \{(\sigma,\tau) : \sigma \lhd \pi \text{ and } \sigma \lhd \tau\}$. Hence

$$\sup_{\boldsymbol{\psi}\lhd\pi} \sup_{\sigma\lhd\boldsymbol{\psi}} \sup_{\tau\rhd\boldsymbol{\psi}} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\tau}}) = \sup_{\sigma\lhd\pi} \sup_{\tau\rhd\sigma} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\tau}}). \tag{115}$$

Again by dominated convergence we have

$$\sup_{\tau\rhd\sigma} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\tau}}) = \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}}). \tag{116}$$

Combining (100), (103), (104), (114), (115), and (116), we have

$$\mathbb{E}\mathbf{1}_C(V_\pi) = \sup_n \sup_{\sigma\lhd\pi} \mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}}). \tag{117}$$

Finally, by Proposition IV.22, the expectation

$$\mathbb{E}q_{n,C}(V_\sigma, V_{\overline{\sigma}}) \tag{118}$$

is a c.e. real relative to χ, uniformly in σ, n, k, m, π, and C. Hence the supremum (117) is a c.e. real relative to χ, uniformly in k, m, π, and C. □

5. Exchangeability in Probabilistic Programs

The computable de Finetti theorem has implications for the semantics of probabilistic functional programming languages, and in particular, gives conditions under which it is possible to remove uses of mutation (i.e., code that modifies a program's internal state). Furthermore, an implementation of the computable de Finetti theorem itself performs this code transformation automatically.

For context, we provide some background on probabilistic functional programming languages. We then describe the code transformation performed by the computable de Finetti theorem, using the example of the Pólya urn and Beta-Bernoulli process discussed earlier. Finally, we discuss partial exchangeability and its role in recent machine learning applications.

5.1. Probabilistic Functional Programming Languages. Functional programming languages with probabilistic choice operators have recently been proposed as universal languages for statistical modeling (e.g., IBAL [Pfe01], $\lambda_\circ$[PPT08], Church [GMR+08], and HANSEI [KS09]). Within domain theory, researchers have considered idealized functional languages that can manipulate exact real numbers, such as Escardó's RealPCF+ [ES99] (based on Plotkin [Plo77]), and functional languages have also been extended by probabilistic choice operators (e.g., by Escardó [Esc09] and Saheb-Djahromi [SD78]).

The semantics of probabilistic programs have been studied extensively in theoretical computer science in the context of randomized algorithms, probabilistic model checking, and other areas. However, the application of probabilistic programs to universal statistical modeling has a somewhat different character from much of the other work on probabilistic programming languages.

In Bayesian analysis, the goal is to use observed data to understand unobserved variables in a probabilistic model. This type of inductive reasoning, from evidence to hypothesis, can be thought of as inferring the hidden states of a program that generates the observed output. One speaks of the *conditional execution* of probabilistic programs, in which they are "run backwards" to sample from the conditional probability distribution given the observed data.

Another important difference from earlier work is the type of algorithms used for conditional inference. Goodman et al. [GMR$^+$08] describe the language Church, which extends a pure subset of Scheme, and whose implementation MIT-Church performs approximate conditional execution via Markov chain Monte Carlo (which can be thought of as a random walk over the execution of a Lisp machine). Park, Pfenning, and Thrun [PPT08] describe the language $\lambda_\circ$, which extends OCaml, and they implement approximate conditional execution by Monte Carlo importance sampling. Ramsey and Pfeffer [RP02] describe a stochastic lambda calculus whose semantics are given by *measure terms*, which support the efficient computation of conditional expectations.

Finally, in nonparametric Bayesian statistics, higher-order distributions (e.g., distributions on distributions, or distributions on trees) arise naturally, and so it is helpful to work in a language that can express these types. Probabilistic functional programming languages are therefore a convenient choice for expressing nonparametric Bayesian statistical models.

The idea of representing distributions by randomized algorithms that produce samples can highlight algorithmic issues. For example, a distribution will, in general, have many different representations as a probabilistic program, each with its own time, space, and entropy complexity. For example, both ways of sampling a Beta-Bernoulli process described in Section 1.1 can be represented in, e.g., the Church probabilistic programming language. One of the questions that motivated the present work was whether there is always an algorithm for sampling from the de Finetti measure when there is an algorithm for sampling the exchangeable sequence. This question was first raised by Roy et al. [RMGT08]. The computable de Finetti theorem answers this question in the affirmative, and, furthermore, shows that one can move

between these representations automatically. In the following section, we provide a concrete example of the representational change made possible by the computable de Finetti transformation, using the syntax of the Church probabilistic programming language.

5.2. Code Transformations. Church extends a pure subset of Scheme (a dialect of Lisp) with a stochastic, binary-valued[2] `flip` procedure, calls to which return independent, Bernoulli($\frac{1}{2}$)-distributed random values in $\{0, 1\}$. Using the semantics of Church, it is possible to associate every closed Church expression (i.e., one without free variables) with a distribution on values. For example, evaluations of the expression

```
(+ (flip) (flip) (flip))
```

produce samples from the Binomial($n = 3$, $p = \frac{1}{2}$) distribution, while evaluations of

```
(λ (x) (if (= 1 (flip)) x 0))
```

always return a procedure, applications of which behave like the probability kernel $x \mapsto \frac{1}{2}(\delta_x + \delta_0)$, where δ_r denotes the Dirac measure concentrated on the real r. Church is call-by-value and so evaluations of

```
(= (flip) (flip))
```

return `true` and `false` with equal probability, while the application of the procedure

```
(λ (x) (= x x))
```

to the argument `(flip)`, written

```
((λ (x) (= x x)) (flip)),
```

always returns `true`. (For more examples, see [GMR+08].)

In Scheme, unlike Church, one can modify the state of a non-local variable using mutation via the `set!` procedure. (In functional programming languages, non-local state may be implemented via other methods. For example, in Haskell, one could use the state monad.) If we consider introducing a `set!` operator to Church, thereby allowing a procedure to modify its environment using mutation, it is not clear that one can associate procedures with probability kernels and closed expressions with distributions. For example, using mutation to maintain and update a counter variable, a procedure could return an increasing sequence of integers on repeated calls. Such a procedure would not correspond with a probability kernel.

[2]The original Church paper defined the `flip` procedure to return `true` or `false`, but it is easy to move between these two definitions.

A generic way to translate code with mutation into code without mutation is to perform a state-passing transformation, where the state is explicitly threaded throughout the program. In particular, a variable representing state is passed into all procedures as an additional argument, transformed in lieu of `set!` operations, and returned alongside the original return values at the end of procedures. Under such a transformation, the procedure in the counter variable example would be transformed into one that accepted the current count and returned the incremented count. One downside of such a transformation is that it obscures conditional independencies in the program, and thus complicates inference from an algorithmic standpoint.

An alternative transformation is made possible by the computable de Finetti theorem, which implies that a particular type of *exchangeable* mutation can be removed *without* requiring a state-passing transformation. Furthermore, this alternative transformation exposes the conditional independencies. The rest of this section describes a concrete example of this alternative transformation, and builds on the mathematical characterization of the Beta-Bernoulli process and the Pólya urn scheme as described in Section 1.1.

Recall that the Pólya urn scheme induces the Beta-Bernoulli process, which can also be described directly as a sequence of independent Bernoulli random variables with a shared parameter sampled from a Beta distribution. In Church it is possible to write code corresponding to both descriptions, but expressing the Pólya urn scheme without the use of mutation requires that we keep track of the counts and thread these values throughout the sequence. If instead we introduce the `set!` operator and track the number of red and black balls by mutating non-local state, we can compactly represent the Pólya urn scheme in a way that mirrors the form of the more direct description using Beta and Bernoulli random variables.

Let a, b be positive computable reals (one can think of them as parameters). We then define `sample-beta-coin` and `sample-pólya-coin` as follows:

(i)

```
(define (sample-beta-coin)
  (let ((weight (beta a b)))
    (λ () (flip weight)) ) )
```

(ii)

```
(define (sample-pólya-coin)
  (let ((red a)
        (total (+ a b)) )
    (λ () (let ((x (flip red/total)))
            (set! red (+ red x))
            (set! total (+ total 1))
            x ) ) )
```

In order to understand these expressions, recall that, given a Church expression E, the evaluation of the `(λ () E )` special form in an environment ρ creates a procedure of no arguments whose application results in the evaluation of the expression E in the environment ρ. The application of either `sample-beta-coin` or `sample-pólya-coin` returns a procedure of no arguments whose application returns (random) binary values. In particular, if we sample two procedures `my-beta-coin` and `my-pólya-coin` via

```
(define my-beta-coin (sample-beta-coin))
(define my-pólya-coin (sample-pólya-coin))
```

then repeated applications of both `my-beta-coin` and `my-pólya-coin` produce random binary sequences that are Beta-Bernoulli processes.

Evaluating `(my-beta-coin)` returns 1 with probability `weight` and 0 otherwise, where the shared `weight` parameter is itself drawn from a Beta(a, b) distribution on $[0, 1]$. The sequence induced by repeated applications of `my-beta-coin` is exchangeable because applications of `flip` return independent samples. Note that the sequence is *not* i.i.d.; for example, an initial sequence of ten 1's would lead one to predict that the next application is more likely to return 1 than 0. However, conditioned on `weight` (a variable hidden within the opaque procedure `my-beta-coin`) the sequence is i.i.d. If we sample another procedure, `my-other-beta-coin`, via

```
(define my-other-beta-coin (sample-beta-coin))
```

then its corresponding `weight` variable will be independent, and so repeated applications will generate a sequence that is independent of that generated by `my-beta-coin`.

The code in *(ii)* implements the Pólya urn scheme with a red balls and b black balls (see [dF75, Chap. 11.4]), and so the sequence of return values from repeated applications of `my-pólya-coin` is exchangeable. Therefore, de Finetti's theorem implies that the distribution of the sequence is equivalent to that induced by i.i.d. draws from the directing random measure. In the case of the Pólya urn scheme, we know that the directing random measure is a random Bernoulli whose parameter has a Beta(a, b) distribution. In fact, the (random) distribution of each sample produced by `my-beta-coin` is such a random Bernoulli. Informally, we can therefore think of `sample-beta-coin` as producing samples from the de Finetti measure of the Beta-Bernoulli process.

Although the distributions on sequences induced by `my-beta-coin` and `my-pólya-coin` are identical, there is an important semantic difference between these two implementations caused by the use of `set!`. While applications of `sample-beta-coin` produce samples from

the de Finetti measure in the sense described above, applications of `sample-pólya-coin` do not; successive applications of `my-pólya-coin` produce samples from different distributions, none of which is the directing random measure for the sequence (a.s.). In particular, the distribution on return values changes each iteration as the sufficient statistics are updated (using the mutation operator `set!`). Perhaps the most obvious difference is that while the sequence produced by repeated applications of `my-pólya-coin` has the same distribution as that produced by repeated applications of `my-beta-coin`, applications of `my-pólya-coin` depend on the non-local state of earlier applications. In contrast, applications of `my-beta-coin` do not depend on non-local state; in particular, the sequence produced by such applications is i.i.d. conditioned on the variable `weight`, which does not change during the course of execution.

The proofs given in this chapter describe an algorithm for computing the de Finetti measure from a representation of the distribution of an exchangeable sequence. We now ask what an implementation of this algorithm would produce in the case of the Beta-Bernoulli example, and more generally.

An implementation of the computable de Finetti theorem (Theorem IV.3), specialized to the case of binary sequences (in which case the de Finetti measure is a distribution on Bernoulli measures and is thus determined by the distribution on $[0, 1]$ of the random probability assigned to the value 1), could transform *(ii)* into a mutation-free procedure whose return values have the same distribution as that of the samples produced by evaluating `(beta a b)`.

In the general case, given a program that generates an exchangeable sequence of reals, an implementation of the computable de Finetti theorem would produce a mutation-free procedure, which we will name `generated-code`, such that applications of the following procedure named `sample-directing-random-measure` and defined by

```
(define (sample-directing-random-measure)
  (let ((shared-randomness (uniform 0 1)))
    (λ () (generated-code shared-randomness)) ) )
```

generate samples from the de Finetti measure in the sense described above. In particular, *(ii)* would be transformed into a procedure `generated-code` such that the sequence produced by repeated applications of the procedure returned by `sample-beta-coin` and the procedure returned by `sample-directing-random-measure` have the same distribution.

In addition to their simpler semantics, mutation-free procedures are often desirable for practical reasons. For example, having sampled the directing random measure, an exchangeable sequence of random variables can be efficiently sampled in parallel without the overhead necessary to communicate sufficient statistics. Mansinghka [Man09] describes some situations where one can exploit conditional independence and exchangeability in probabilistic programming languages for improved parallel execution.

5.3. Partial Exchangeability of Arrays and Other Structures. The example above involved binary sequences, but the computable de Finetti theorem can be used to transform implementations of real exchangeable sequences. Consider the following exchangeable sequence whose combinatorial structure is known as the Chinese restaurant process (see Aldous [Ald85]). Let $\alpha > 0$ be a computable real and let H be a computable distribution on $\mathbb{R}$. For $n \geq 1$, each X_n is sampled in turn according to the conditional distribution

$$\mathbf{P}[X_{n+1} \mid X_1, \ldots, X_n] = \frac{1}{n+\alpha}\Big(\alpha\, H + \sum_{i=1}^{n} \delta_{X_i}\Big). \tag{119}$$

The sequence $\{X_n\}_{n\geq 1}$ is exchangeable and the directing random measure is a Dirichlet process whose "base measure" is αH. Given such a program, we can automatically recover the underlying Dirichlet process prior, samples from which are random measures whose discrete structure was characterized by Sethuraman's "stick-breaking construction" [Set94]. Note that the random measure is not produced in the same manner as Sethuraman's construction and certainly is not of closed form. But the resulting mathematical objects have the same structure and distribution.

Exchangeable sequences of random objects other than reals can often be given de Finetti-type representations. For example, the Indian buffet process, defined by Griffiths and Ghahramani [GG05], is the combinatorial process underlying a *set-valued* exchangeable sequence that can be written in a way analogous to the Pólya urn in *(ii)*. Just as the Chinese restaurant process gives rise to the Dirichlet process, the Indian buffet process gives rise to the Beta process (see Thibaux and Jordan [TJ07] for more details).

In the case where the "base measure" of the underlying Beta process is *discrete*, the resulting exchangeable sequence of sets can be transformed into an exchangeable sequence of *integer* indices (encoding finite multisets of the countable support of the discrete base measure). If

we are given such a representation, the computable de Finetti theorem implies the existence of a computable de Finetti measure.

However, the case of a continuous base measure is more complicated. Unlike in the Chinese restaurant process example, which was a sequence of random reals, the computable de Finetti theorem is not directly applicable to exchangeable sequences of random sets, although there is an embedding such that a version of the computable de Finetti theorem for computable Polish spaces might suffice. A "stick-breaking construction" of the Indian buffet process given by Teh, Görür, and Ghahramani [TGG07] is analogous to the code in *(i)*, but samples only a Δ_1-index for the (a.s. finite) sets, rather than a canonical index (see Soare [Soa87, II.2]); however, many applications depend on having a canonical index. This observation was first noted by Roy et al. [RMGT08]. Similar problems arise when using the Inverse Lévy Measure method [WI98]. The computability of the de Finetti measure in the continuous case remains open.

Combinatorial structures other than sequences have been given de Finetti-type representational theorems based on notions of *partial* exchangeability. For example, an array of random variables is called *separately* (or *jointly*) exchangeable when its distribution is invariant under (simultaneous) permutations of the rows and columns and their higher-dimensional analogues. Nearly fifty years after de Finetti's result, Aldous [Ald81] and Hoover [Hoo79] showed that the entries of an infinite array satisfying either separate or joint exchangeability are conditionally i.i.d. These results have been connected with the theory of graph limits by Diaconis and Janson [DJ08] and Austin [Aus08] by considering the adjacency matrix of an exchangeable random graph.

As we have seen with the Beta-Bernoulli process and other examples, structured probabilistic models can often be represented in multiple ways, each with its own advantages (e.g., representational simplicity, compositionality, inherent parallelism, etc.). Extensions of the computable de Finetti theorem to partially exchangeable settings could provide analogous transformations between representations on a wider range of data structures, including many that are increasingly used in practice. For example, the Infinite Relational Model [KTG$^+$06] can be viewed as an urn scheme for a partially exchangeable array, while the hierarchical stochastic block model constructed from a Mondrian process in [RT09] is described in a way that mirrors the Aldous-Hoover representation, making the conditional independence explicit.

6. Predictive distributions and posterior analysis in exchangeable sequences

While conditioning is not computable in the general case, we can sometimes exploit additional structure to compute conditional distributions. In this section, we continue our study of exchangeable sequences, but from the perspective of computing conditional distributions of directing random measures. In particular, for an exchangeable sequence $\{X_k\}_{k\in\mathbb{N}}$ with directing random measure ν, we show that there is an algorithm for computing the posterior distributions $\{\mathbf{P}[\nu|X_{1:k}]\}_{k\geq 1}$ if and only if there is an algorithm for sampling from the predictive distributions $\{\mathbf{P}[X_{k+1}|X_{1:k}]\}_{k\geq 1}$.

Note that ν is, in general, an infinite dimensional object. However, in many settings, the directing random measure corresponds to a particular member of a parametrized family of distributions, and in this case, the de Finetti measure corresponds to a distribution on parameters. Note that while the de Finetti measure is often interpreted as a prior in the Bayesian setting, it is uniquely determined by the distribution of the exchangeable sequence, which itself may be described without reference to any such prior.

Example IV.24. Consider the sequence $\{Y_k\}_{k\geq 1}$ where, for each $k \geq 1$, the conditional distribution of Y_k given $Y_1, \ldots, Y_{k-1}$ is normally distributed with mean $\frac{1}{k}\sum_{i=1}^{k-1} Y_i$ and variance $1 + \frac{1}{k}$. The sequence $\{Y_k\}_{k\geq 1}$ can be shown to be exchangeable and its directing random measure is a random Gaussian with unit variance but random mean, and so each realization of the directing random measure is associated with (and completely characterized by) a corresponding mean parameter. Let Z be the mean of the directing random measure. The sequence $\{Y_k\}_{k\geq 1}$ is conditionally i.i.d. given Z. Furthermore, it can be shown that the distribution $\mathbf{P}_Z$ of Z is a standard normal distribution. The de Finetti measure can be derived from $\mathbf{P}_Z$, as follows: Let $M : \mathbb{R} \to \mathcal{M}_1(\mathbb{R})$ be the map that takes a real m to the Gaussian distribution with mean m and unit variance. Then the de Finetti measure μ is given by $\mu(B) = \mathbf{P}_Z(M^{-1}(B))$, where B is a (Borel measurable) subset of $\mathcal{M}_1(\mathbb{R})$ and $M^{-1}(B)$ is the inverse image of B under the map M. In summary, while a random Gaussian distribution renders the sequence conditionally i.i.d., the latent mean parameter Z of the random Gaussian captures the structure of the sequence.

Remark IV.25. Recall the Dirichlet process example in Chapter IV. The computable de Finetti's theorem shows that an exchangeable sequence is computable if and only if its de Finetti measure is computable,

and in this example, the computability of the Blackwell-MacQueen urn scheme implies the computability of the Dirichlet process prior. While the most common way of sampling a Dirichlet process is via the stick breaking representation given by Sethuraman [Set94], we should not expect the output of the computable de Finetti theorem to make any use of this representation (just as it would not identify the mean of the random Gaussian as the one-dimensional quantity of interest in the Gaussian case).

Recall that the stick breaking representation is the list of atoms (and their masses) that comprise the Dirichlet process. We note the following fact about computably recovering this representation from a discrete computable probability measure. First, recall the elementary computability fact that the equality of two computable reals is only co-semidecidable, meaning that we can eventually recognize two reals are unequal, but cannot in general recognize when they are the same. Likewise, given a representation of a distribution that is known to be discrete, although we can compute a list of the atoms (as a point in $\mathbb{R}^\infty$), it is not possible to recover their masses. Therefore, the function taking a discrete measure to its stick breaking representation is not computable. Thus, even in settings where the computable de Finetti theorem tells us that the directing random measure is computably distributed, it may (as in Example IV.24) or may not (as described here) be possible to transform the random measure into another random variable that also renders the sequence conditionally i.i.d. Of course, the stick breaking prior is obviously computable; we simply cannot expect it to read it off from output we receive from the computable de Finetti theorem.

6.1. Posterior analysis in exchangeable sequences. Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence of real-valued random variables. Even if the distribution of X is computable, $\mathbf{P}[X_{k+1}|X_{1:k}]$ is not necessarily computable. To see this, let ν_0 and ν_1 be distributions concentrated on the rationals and irrationals such that the function $w : \{\nu_0, \nu_1\} \subseteq \mathcal{M}_1(\mathbb{R}) \to \{0, 1\}$ given by $w(\nu_i) = i$ is computable (i.e., we can computably distinguish the distributions). Let ν be a random measure which with equal probability is either ν_0 or ν_1 and let X_i be a conditionally i.i.d. sequence with directing random measure ν. Then

$$\mathbf{P}[X_2|X_1] = \begin{cases} \nu_0, & \text{if } X_1 \text{ rational} \\ \nu_1, & \text{if } X_1 \text{ irrational} \end{cases} \qquad \text{a.s.} \tag{120}$$

If some version of the conditional distribution were computable, we would be able to decide rationality on $\mathbb{R}$, which is undecidable. Hence, no version is computable.[3]

However, in most cases, our knowledge of an exchangeable sequence is, in fact, precisely of this form: an algorithm (or *predictive rule*) which, given samples for a prefix $X_{1:k}$, describes the conditional distribution of the next element, X_{k+1}. By induction, we can use the predictive rule to subsequently sample from the conditional distribution of X_{k+2} given $X_{1:k+1}$, and so on, sampling an entire infinite exchangeable sequence given the original prefix. The following result shows that the ability to sample consistently (i.e., from the true posterior predictive) is equivalent to being able to compute the posterior distribution of the latent distribution that is generating the sequence.

Theorem IV.26. *Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence of random variables with directing random measure ν. There is a program that, given a representation of the sequence of posterior predictives $\{\mathbf{P}[X_{k+1}|X_{1:k}]\}_{k\geq 0}$, outputs a representation of the sequence of posterior distributions $\{\mathbf{P}[\nu|X_{1:k}]\}_{k\geq 0}$, and vice-versa.*

Proof sketch. Suppose we are given (a representation of) $\{\mathbf{P}[\nu|X_{1:k}]\}_{k\geq 0}$. Fix $j \geq 0$ and an observation $x_{1:j} \in \mathbb{R}^j$. We can compute (a representation of) $\mathbf{P}[X_{j+1}|X_{1:j}]$ by computing samples from the distribution $\mathbf{P}[X_{j+1}|X_{1:j} = x_{1:j}]$, given $x_{1:j}$. But by assumption, we can sample $\hat{\nu} \sim \mathbf{P}[\nu|X_{1:j} = x_{1:j}]$, and then sample $\hat{X}_{k+1} \sim \hat{\nu}$.

To prove the converse, fix $j \geq 0$ and observe that, conditioned on $X_{1:j}$, the sequence $\{X_{j+1},\ X_{j+2},\ \dots\}$ is an exchangeable sequence whose de Finetti measure is $\mathbf{P}[\nu|X_{1:j}]$. We show how to compute the conditional distribution of this exchangeable sequence, and then invoke the computable de Finetti theorem to compute the posterior $\mathbf{P}[\nu|X_{1:j}]$.

Suppose we are given (a representation of) $\{\mathbf{P}[X_{k+1}|X_{1:k}]\}_{k\geq 0}$. Given observed values $x_{1:j}$ for a prefix $X_{1:j}$, we can sample

$$\hat{X}_{j+1} \sim \mathbf{P}[X_{k+1}|X_{1:k} = x_{1:k}]. \tag{121}$$

Then, treating $\{x_{1:j}, \hat{X}_{j+1}\}$ as observed values for $X_{1:j+1}$, we can sample $\hat{X}_{j+2} \sim \mathbf{P}[X_{j+2}|X_{1:j+1} = \{x_{1:j}, \hat{X}_{j+1}\}]$.

By an inductive argument, we can therefore sample from the conditional distribution of the exchangeable sequence $X_{j+1:\infty}$ given $X_{1:j} =$

[3]Note that every version is discontinuous on every measure one set, like the discontinuous counterexample to computability in Chapter III. It may be possible to construct an exchangeable sequence whose predictive and posterior distributions are almost continuous despite being noncomputable using the computable random variables (N, X) developed in the same chapter.

$x_{1:j}$, and so we can compute the conditional distribution of the exchangeable sequence.

Finally, by the Computable de Finetti Theorem (Chapter IV, Theorem IV.3), we can compute the de Finetti measure, $\mathbf{P}[\nu|X_{1:k}]$, from the distribution of the conditionally exchangeable sequence $X_{j+1:\infty}$. □

Note that the "natural" object here is the directing random measure ν itself, which is not necessarily the natural parameter Θ for which $\nu = \mathbf{P}[X_1|\Theta]$. While a particular parametrization may be classically unidentifiable or noncomputable, the directing random measure is always identifiable and computable.

The hypothesis of Theorem IV.26 captures a common setting in nonparametric modeling, where a model is given by a prediction rule. Such representations can exist even when there is no Bayes' rule.

Example IV.27. Recall the Dirichlet process example in Chapter IV. Note that the Blackwell-MacQueen prediction rule satisfies the hypotheses of Theorem IV.26. The proof of Theorem IV.26 (if implemented as code) automatically transforms the prediction rule into the (computable) posterior distribution

$$\{x_i\}_{i\leq k} \mapsto \mathrm{DP}(\alpha H + \textstyle\sum_{i=1}^{k} \delta_{x_i}). \tag{122}$$

Posterior computation for many other species sampling models [Pit96] is likewise possible because these models are generally given by computable predictive distributions. As another example, exact posterior analysis for traditional Pólya trees, a flexible class of random distributions, is possible. In contrast, nearly all existing inference techniques for Pólya trees make truncation-based approximations. For arbitrary Pólya trees, the noncomputability result implies that there is no algorithm that can determine the error introduced by a given truncation.

Note that, trivially, any model for which someone has constructed an exact posterior algorithm necessarily has a computable predictive, and so the hypotheses of the algorithm are quite general. In particular, note that exchangeable structure need not be evident for Theorem IV.26 to apply. Let G be a distribution on $\mathcal{M}_1(\mathbb{R})$; sample $F \sim G$, and then sample an observation $X \sim F$ from the random distribution F. Can we compute $\mathbf{P}[F|X]$? Theorem IV.26 implies that if we introduce nuisance variables $X_2, X_3, \ldots$ that are themselves independent draws from F, then $\mathbf{P}[F|X]$ is computable if the sequence $\mathbf{P}[X_2|X], \mathbf{P}[X_3|X, X_2], \ldots$ is computable. So even though the model only invokes a single sample from F, the ability to do posterior analysis on F given X is linked to our ability to sample the sequence $X_2, X_3, \ldots$ given X.

6.2. Related work. Orbanz [Orb10] proves a version of Kolmogorov's extension theorem for families of conditional distributions, providing a way to construct nonparametric Bayesian models. In particular, Orbanz shows how to construct a (countable-dimensional) nonparametric model as the limit of a conditionally projective family of finite dimensional conditional distributions, and shows that the limiting nonparametric prior will be conjugate exactly when the projective family is.

Essentially, in order to obtain a *closed form* expression (in terms of sufficient statistics) for the posterior of a nonparametric model, one must construct the nonparametric model as the projective limit of models that admit both sufficient statistics and a conjugate posterior (the main examples of which are the projective limits of exponential family models).

We now give a related statement: in order to *computably* recover the posterior distribution from sufficient statistics of the observations, it is necessary and sufficient to be able to computably sample new observations given sufficient statistics of past observations.

For simplicity, we restrict our attention to sufficient statistics of the form $\sum_{i=1}^{k} T(X_i)$, where $T : \mathbb{R} \to \mathbb{R}^m$ is a continuous function. This setting covers essentially all natural exponential family likelihoods.

When the sufficient statistic and the conditional distributions

$$\mathbf{P}[X_{k+1} | \textstyle\sum_{i=1}^{k} T(X_i)], \tag{123}$$

for $k \geq 1$, are computable (and hence their composition is a computable predictive distribution), we get as an immediate corollary that we can compute the posterior from the sufficient statistic, and therefore, the sufficiency for the predictive carries over to the posterior.

Corollary IV.28. *Let X and ν be as above, and let $\sum_{i=1}^{k} T(X_i)$ for $T : \mathbb{R} \to \mathbb{R}^m$ be a sufficient statistic for X_{k+1} given $X_{1:k}$. Then the sequence of posterior distributions $\mathbf{P}[\nu | \sum_{i=1}^{k} T(X_i)]$ for $k \geq 1$ is computable if and only if the sequence of conditional distributions $\mathbf{P}[X_{k+1} | \sum_{i=1}^{k} T(X_i)]$, for $k \geq 1$, and the sufficient statistic T are computable.*

Corollary IV.28 and Theorem IV.26 provide a framework for explaining why ad-hoc methods for computing conditional distributions have been successful in the past, even though the general task is not computable.

The classical focus on closed form solutions has necessarily steered the field into studying a narrow and highly constrained subspace of computable distributions. The class of computable distributions includes

many objects for which we cannot find (or for which there does not even exist) a closed form. On the other hand, computable distributions provide, by definition, a mechanism for computing numerical answers to any desired accuracy.

Massive computational power gives us the freedom to seek more flexible model classes. Armed with general inference algorithms and the knowledge of fundamental limitations, we may begin to explore new frontiers along the interface of computation and statistics.

As a practical matter, the algorithm (given by the proof of the Computable de Finetti Theorem) that transforms from the law of an exchangeable sequence to its de Finetti measure can be very inefficient. It is an open challenge to identify circumstances where the de Finetti measure can be computed efficiently. There are many intriguing nonuniform questions as well: do efficiently samplable exchangeable sequences have efficiently samplable de Finetti measures? Do efficiently samplable exchangeable sequences have efficiently samplable predictive rules?

CHAPTER V

Distributions on data structures: a case study

Data structures are foremost a representation of knowledge, either of a program's own state or of the state of the world with which a program is interacting. While the study of data structures can be traced back to the early days of the field of computer science, data structures, and in particular *random* data structures representing uncertain structure in the real world, have a new and important role to play in probabilistic programming and machine learning more generally.

From a Bayesian point of view, the way to discover (and subsequently take advantage of) structure in data is to first define a hypothesis space $\mathcal{H}$ of possible structures and then place a distribution on the joint space of hypotheses and observable data, encoding one's beliefs about the relative likelihoods of an unobserved structure and observed data. The conditional distribution on $\mathcal{H}$ given actual data encodes our updated beliefs as to the relative likelihoods of competing structures.

When such models are implemented in actual systems, the abstract structures in $\mathcal{H}$ often become literal instances of data structures and, in these cases, the abstract distribution on $\mathcal{H}$ can be interpreted as a distribution on data structures. This connection is even stronger in the probabilistic programming setting, wherein a structured probabilistic model is specified by a probabilistic program that literally constructs a random data structure in the course of generating hypothesized data.

The trend in Bayesian statistics towards the use of nonparametric models can be understood as a shift in the types of data structures and other abstractions one finds in the corresponding probabilistic programs. Classic finite-dimensional models (e.g., linear regression) can be implemented with simple array data structures to store model parameters and primitive forms of recursion to generate a data set. In contrast, the burgeoning field of Bayesian nonparametric statistics has at its core a number of stochastic processes (e.g., Dirichlet process) that are best described using unbounded recursion, and implemented using linked lists, streams and higher-order procedures. A similar observation was made by Jordan [Jor10].

A central thesis of my work is that the probabilistic programming perspective can vastly simplify the ever-expanding landscape of Bayesian nonparametric statistics. In particular, many nonparametric constructs have simple interpretations in suitably flexible probabilistic programming languages. One example is the Dirichlet process, which from a probabilistic programming perspective can be viewed as performing a stochastic variant of memoization [GMR+08, RMGT08]. The usefulness of this perspective has been demonstrated by O'Donnell [O'D11], who significantly expands on the idea of stochastic memoization in order to define state of the art models of productivity and reuse in natural language.

As we search for new nonparametric processes, we can use the probabilistic programming perspective to guide us. This chapter presents a case study where the problem of placing an exchangeable distribution on an array of random variables leads to the problem of defining a distribution on the space of infinite kd-tree data structures.

1. Exchangeable arrays

Relational data sets capture the interactions *between* objects. For example, hyperlinks are a relation on pairs of websites, and large data sets capturing the relational hyperlink structure of the web are critical to modern Internet search algorithms. Other examples of (binary) relations include friendship in social networks, and movie ratings by users. A natural way to represent a relation between sets of objects is as an array. In order to build probabilistic models of relational data, we will therefore be interested in arrays of random variables.

The notion of exchangeability plays a central role in hierarchical Bayesian modeling of sequences. Related notions of symmetry are central in the relational setting: *a priori*, our uncertainty about an array representing a relation may be invariant to the ordering of the rows and columns—i.e., the particular way we index into the objects. In other words, a natural notion of exchangeability for relational data is that objects (not individual relationships) are exchangeable.

The theory of exchangeable arrays developed by Aldous [Ald81] and Hoover [Hoo79], and extended by Kallenberg (see [Kal05, Sec. 7.6] for details) gives a natural representation for exchangeable relational data. Roughly speaking, each object is represented by a latent variable taking values in a latent space and a relation is a random function on the product of the latent spaces. The relation between two objects is simply the value of the function evaluated at the latent representations of the two objects.

Formally, we say that an infinite array $(R_{i,j})_{i,j\geq 1}$ is **separately exchangeable** if its distribution is invariant to separate permutations on its rows and columns. That is, for all $n, m \geq 1$ and permutations $\pi \in S_n$ and $\sigma \in S_m$, we have

$$(R_{i,j} \,:\, i \leq n,\ j \leq m) \stackrel{d}{=} (R_{\pi(i),\sigma(j)} \,:\, i \leq n,\ j \leq m). \tag{124}$$

Aldous [Ald81] and Hoover [Hoo79] showed that all separately exchangeable arrays taking values in a space S satisfy

$$R_{i,j} = f(\theta, \xi_i, \eta_j, \delta_{i,j}) \quad a.s. \tag{125}$$

for some measurable function $f : [0,1]^4 \to S$ and independent uniformly distributed random variables $\theta, (\xi_i), (\eta_j), (\delta_{i,j})$, $i, j \in \mathbb{N}$. In order to understand this representational result, we will study the special case $S = \{0, 1\}$ of binary-valued binary relations.

Let

$$h(t, x, y) := \mathbf{P}\{f(t, x, y, \delta) = 1\}, \tag{126}$$

where $\delta \sim U[0,1]$, and define $H_\theta(x, y) := h(\theta, x, y)$, i.e., H_θ is a random function from $[0,1]^2 \to [0,1]$. Then the entries $(R_{i,j})$ are mutually independent given $H_\theta, (\xi_i), (\eta_j)$, $i, j \in \mathbb{N}$, and

$$\mathbf{P}[R_{i,j} = 1 \mid H_\theta, \xi_i, \eta_j] = H_\theta(\xi_i, \eta_j). \tag{127}$$

Put simply, the relationship between object i and j is conditionally independent of all other relationships given the "link" function $H := H_\theta$ and "representations" ξ_i and η_j for each column i and row j, respectively. Therefore, a separately exchangeable distribution on an array encodes a prior distribution on the unknown representations of rows and columns, as well as a prior distribution on the unknown link function.

1.1. Bayesian models of exchangeable arrays. Just as the de Finetti representation theorem serves as the theoretical foundation for hierarchical Bayesian models of exchangeable sequences, the Aldous-Hoover representation theorem serves as the foundation for models of exchangeable relational data (see [Hof08] and [RT09] for further discussion). In particular, the central problem of modeling an exchangeable binary-valued array is choosing an appropriate function space and prior distribution for the link function H. (Note that the choice of a uniform random variable as a representation is not essential as any random element ξ on a Borel space satisfies $\xi = f(\theta)$ a.s. for some uniform random variable θ and measurable function f.)

One of the most popular probabilistic models for relational data is the *stochastic block model*. Stochastic block models were introduced by Holland, Laskey and Leinhardt [HLL83], although more recent

incarnations (see [KTG+06, XTYK06, RKMT07]) descend from work by Waserman and Anderson [WA87] and Nowicki and Snijders [NS01]. At a high level, stochastic block models assume that every object has a *type* and that each relationship is determined (up to independent noise) by the types of objects interacting, not the identity of the individual objects. For example, a crude model of politics might posit that the relationship between two members of Congress is characterized by their party membership. Stochastic block models can be seen as an extension of the paradigm of *clustering* to the relational setting.

A formal description of a particular stochastic block model called the Infinite Relational Model [KTG+06, XTYK06] (henceforth, IRM) reveals a shortcoming that we will attempt to address. Note that following presentation is specialized to the binary case.) Let $\delta_1 > \delta_2 > \cdots > 0$ denote the jump times of a nonhomogeneous Poisson process on $(0, 1]$ whose rate measure is the so-called *beta Lévy measure* $\mu(dx) = \alpha x^{-1} dx$, and put $\delta_0 = 1$. (See [Kin93] for an excellent introduction to nonhomogeneous Poisson processes.) Teh, Görür, and Ghahramani [TGG07] show that the inter-jump wait times $\delta_n - \delta_{n+1}$ have the same distribution as the "stick lengths" in the stick-breaking construction of the Dirichlet process [Set94].

Let (δ'_n) be an independent copy of (δ_n), and let $(\theta_{i,j})$ be an i.i.d. array of Beta random variables. Then the random link function H corresponding to an IRM distribution is given by

$$H(x, y) = \theta_{n,m}, \tag{128}$$

where n, m are the unique indices such that $x \in (\delta_n, \delta_{n-1}]$ and $y \in (\delta'_m, \delta'_{m-1}]$. In particular, the entries $(R_{i,j})$ of an array with an IRM distribution are conditionally independent and satisfy (127) for some pair of independent sequences (ξ_i) and (η_j) of i.i.d. uniform random variables.

It is clear from (128) that the row and column objects are first separately and independently clustered, and that each pair (n, m) of clusters has a prototypical probability $\theta_{n,m}$ of interacting. Therefore, stochastic block models can been seen to place a distribution on a partition of the product space by forming the product of the partitions on the row and column space. The problem with such an approach is that there is rarely a single partition of the objects into types that accounts for the variability of the data. One can visualize this drawback: the "resolution" needed to model fine detail in one area of the array necessarily causes other parts of the array to be dissected into unnecessarily distinct clusters, even if the data suggest there is no such structure (see Figure 1).

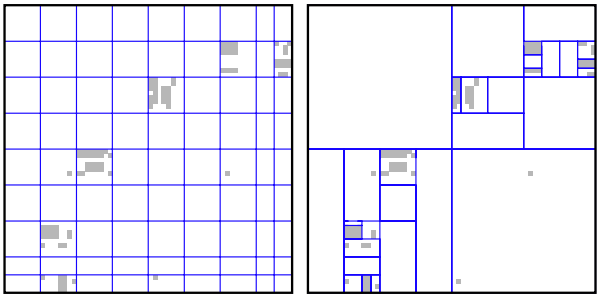

FIGURE 1. (left) Under a suitable permutation of the rows and columns, stochastic block models like the IRM [KTG+06] induce a regular partition on the product space, introducing structure where the data do not support it. (right) More flexible partitions, such as guillotine partitions, can provide resolution where it is needed.

2. Random kd-trees

The approach we will take is to consider a richer class of functions corresponding to a richer set of partitions, and in particular, nested, axis-aligned, binary partitions of a space. Within computer science, such structures are known as multidimensional binary search trees, or simply, kd-trees [Ben75]. Within combinatorics and other areas of mathematics, such structures are known as guillotine partitions.

A kd-tree data structure can be defined inductively as either 1) a leaf (representing the trivial partition where all points are in the same equivalence class) or 2) an axis-aligned cut, composed of two children kd-tree data structures.

Given the recursive definition of kd-trees, it is natural to consider distributions corresponding with probabilistic programs of the form:

SAMPLEP(D), $D \subseteq \mathbb{R}^D$

1. **with some probability**
2. **return** empty-leaf.
3. **otherwise**
4. Sample an axis-aligned partition $\{D_0, D_1\}$ of D
5. **return** $\langle c, \text{SAMPLEP}(D_0), \text{SAMPLEP}(D_1)\rangle$.

Does a process with this structure produce distributions with desirable properties? With what probability should the procedure stop and return a leaf? If a random cut is made, from what distribution should a cut be drawn? These choices determine the properties of the resulting distribution.

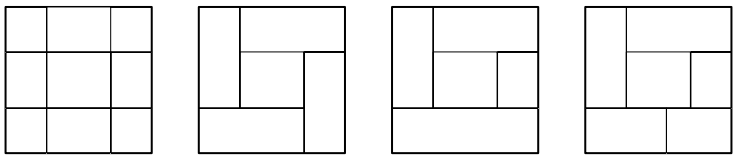

FIGURE 2. (a-d) Floorplan partitions of the unit square. (c) and (d) are also guillotine partitions. (d) is an atomic refinement of (c).

A process of the above form makes several ontological assumptions that deserve consideration. In particular, assuming that the procedure halts with probability one, it will always return finite partitions, and moreover, the choice of the stopping probabilities will implicitly define a distribution on the size and shape of the partition. However, a finite partition may be an inappropriate assumption when the amount of data we wish to model is potentially unbounded.

In order to consider partitions of potentially unbounded complexity, we will need a way to generate infinite partitions. The key tool at our disposal are projective limits and in order for these methods to be applicable, we will need a procedure with the following key property: that it should produce distributions on partitions that are invariant under the operation of restricting the partition to a subset. That is, if we partition a space X and then look at the induced partition on a subset $A \subseteq X$, this distribution should match the distribution that we would have produced starting from A directly.

In the remainder of the chapter we present a construction of a random guillotine partition that achieves this property.

3. Guillotine partitions and Mondrian processes

The basic constituent of the partitions we will be considering is an axis-aligned cuboid or simply **box**. In particular, a box is a subset $A \subseteq \mathbb{R}^D$ of the form $A = I_1 \times \cdots \times I_D$ for left-open/right-closed intervals $I_d \in \mathbb{R}$, i.e., $I_d \in \{(-\infty, b], (a, b], (a, \infty) : a, b \in \mathbb{R},\ a < b\}$. Note that the whole of $\mathbb{R}^D$ is considered a box by this definition.

Let X be a box in $\mathbb{R}^D$. A **floorplan partition (of X)** is a partition of X into disjoint boxes. We denote the set of all floorplan partitions of X by $\mathscr{F}_X$. The set of floorplan partitions and its combinatorial structure is the subject of a large literature (see, e.g., [Sto83] and [WL89]).

Let π be a floorplan partition of a box X. We call a floorplan partition $\eta \in \mathscr{F}_X$ a **refinement of** π, written $\eta \succeq \pi$, when $\eta = \bigcup_{A \in \pi} \beta_A$ for some floorplan partitions $\beta_A \in \mathscr{F}_A$. It is straightforward to show

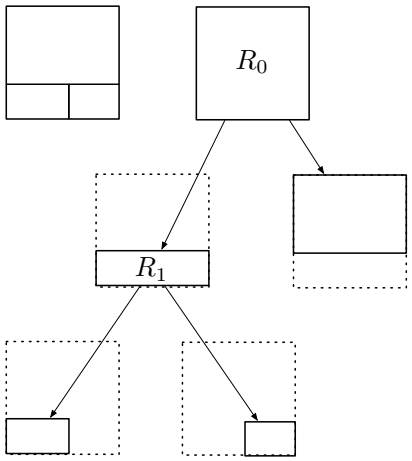

FIGURE 3. The structure of a guillotine partition can be represented by a rooted hierarchy of cuts. Note that this identification is not unique; the partition $\boxplus$ can be achieved starting with a vertical or horizontal cut.

that the relation $\succeq$ is a complete partial order on $\mathscr{F}_X$. We write the irreflexive relation $\eta \succ \pi$ when $\eta \neq \pi$ and $\eta \succeq \pi$.

We call η an **atomic** refinement of π, written $\eta \succ_1 \pi$, when $\eta \succ \pi$ and $|\eta \setminus \pi| = 2$ (see Figure 2d) and write $\eta \succeq_1 \pi$ when $\eta \succ_1 \pi$ or $\eta = \pi$. An atomic refinement of a floorplan partition is generated by "cutting" a box into two subboxes. The set of partitions generated by sequences of cuts are known as guillotine partitions:

Definition V.1 (Finite guillotine partition)**.** The set of **finite guillotine partitions on** X is the smallest set $\mathscr{G}^0_X$, such that

(1) $\{X\} \in \mathscr{G}^0_X$; and
(2) $\eta \in \mathscr{F}_X$, $\pi \in \mathscr{G}^0_X$, $\eta \succ_1 \pi$ implies $\eta \in \mathscr{G}^0_X$.

That is, $\mathscr{G}^0_X$ is the transitive closure of $\{X\}$ under $\succeq_1$. We define $\mathscr{G}^0 := \bigcup_X \mathscr{G}^0_X$, where the union is over subboxes $X \subseteq \mathbb{R}^D$. (Like floorplan partitions, guillotine partitions, and in particular their combinatorial structure, have been the subject of much research. See, e.g., [GZ89] and [AM10].)

Guillotine partitions can be seen to have hierarchical structure. Let π be a finite guillotine partition. Then there is at least one finite sequence of atomic refinements $\pi = \pi_k \succ_1 \pi_{k-1} \succ_1 \cdots \succ_1 \pi_1 \succ_1 \pi_0 = \{X\}$ connecting π and the trivial partition $\{X\}$, each resulting from a cut to some box creating two sub-boxes. These refinements can be seen to form a rooted, binary tree. In particular, let $R_i \in \pi_i \setminus \pi_{i+1}$ denote the box cut on the ith step, and associate each such set R_i with children

$\pi_{i+1} \setminus \pi_i$. Then the $\{R_i\}$ comprise the internal nodes of a binary tree with root R_0 and leaves $A \in \pi$ (see Figure 3).

A partition of a set naturally induces a partition on any subset. To make this notion precise, we define:

Definition V.2 (Restriction)**.** Let $A \subseteq X$ and let $\pi \subseteq \mathcal{P}(X) \setminus \{\emptyset\}$. We call the subset of $\mathcal{P}(A)$ given by

$$\Pi_A \pi := \{A \cap B : B \in \pi\} \setminus \{\emptyset\} \tag{129}$$

the **restriction of π to A**.

It is straightforward to show that a restriction of a partition is again a partition. As the intersection of two overlapping boxes is again a box, it follows easily that floorplan partitions are closed under restriction to subboxes, i.e., the restriction of a floorplan partition to a subbox is again a floorplan partition of that subbox. The same is true of guillotine partitions:

Lemma V.3 (Closed under restriction)**.** *Let X be a box and $\pi \in \mathscr{G}_X^0$ a finite guillotine partition. For all subboxes $A \subseteq X$, we have that $\Pi_A \pi \in \mathscr{G}_A^0$. In particular, $\mathscr{G}_A^0 = \{\Pi_A \pi : \pi \in \mathscr{G}_X^0\}$.*

Proof. By the definition of finite guillotine partitions, it is the case that $\pi = \pi_k \succ_1 \pi_{k-1} \succ_1 \cdots \succ_1 \pi_1 \succ_1 \pi_0 = \{X\}$, for some finite sequence of guillotine partitions (π_n). Note that $\eta \succ_1 \nu$ implies that $\Pi_A \eta \succeq_1 \Pi_A \nu$. Therefore, $\Pi_A \pi = \Pi_A \pi_k \succeq_1 \cdots \succeq_1 \Pi_A \pi_0 = \Pi_A \{X\} = \{A\}$, hence $\Pi_A \pi \in \mathscr{G}_A^0$. □

Many useful operations on sets and partitions can be described by element-wise transformations. For a function $f : X \to Y$, we write f‘ to denote the function from $\mathcal{P}(X) \to \mathcal{P}(Y)$ given by

$$f\text{‘}(A) = \{f(x) : x \in A\}, \qquad A \subseteq X. \tag{130}$$

We will write f“ to denote $(f\text{‘})\text{‘}$, i.e., the operator is left associative.[1]

As an example, consider the translation $\tau(x) = x + v$, where $v \in \mathbb{R}^D$. Then τ‘ takes sets in $\mathbb{R}^D$ to sets in $\mathbb{R}^D$. In particular, $\tau\text{‘}A = \{v + x : x \in A\}$ for $A \subseteq \mathbb{R}^D$; informally, A is translated by v. It follows that τ“ acts on sets of sets (and in particular, partitions), translating each constituent set by v. E.g., if π is a partition of a box X, then $\tau\text{“}\pi$ is a partition of the box $\tau\text{‘}X$. As another example, Lemma V.3 shows that $\Pi_Y\text{‘}\mathscr{G}_X^0 = \mathscr{G}_Y^0$.

[1]We apologize to set theorists who use double apostrophes “ to denote the operator that we have chosen to represent by a single apostrophe ‘. However, ““ seemed cumbersome.

3.1. Random guillotine cut. Let X be a box, let $\pi \in \mathscr{G}_X^0$ be a guillotine partition of X, and let

$$R_\pi := \{\eta \in \mathscr{G}_X^0 \,:\, \eta \succ_1 \pi\} \tag{131}$$

be the set of atomic refinements of π. We would like to define a "uniform" distribution on R_π.

There is no standard notion of a uniformly distributed refinement, and it is not hard to generate distinct distributions on R_π that could all be justifiably called uniform. We will take an axiomatic approach, and instead consider consistency properties on an entire family

$$\{\nu_\pi \,:\, \pi \in \mathscr{G}^0\} \tag{132}$$

of measures on atomic refinements. In particular, we will restrict our attention to families such that, for all $\pi \in \mathscr{G}^0$, we have invariance under:

(1) **translation**, i.e., $\nu_\pi = \nu_{\tau\text{“}\pi} \circ \tau$“, for all $\tau(x) = x + v$ and $v \in \mathbb{R}^D$;
(2) **permutation (of dimensions)**, i.e. $\nu_\pi = \nu_{\rho\text{“}\pi} \circ \rho$“, for all $\rho(y_1, \ldots, y_D) = (y_{\sigma(1)}, \ldots, y_{\sigma(D)})$ and permutations σ of $\{1, \ldots, D\}$; and finally
(3) **restriction**, i.e., $\nu_\pi(A) = (\nu_{\Pi_Y \pi} \circ \Pi_Y$‘$)(A)$, for all subboxes $Y \subseteq X$ and measurable subsets $A \subseteq \{\xi \in R_\pi \,:\, \Pi_Y \xi \neq \Pi_Y \pi\}$.

In fact, as we will show, these three invariances pin down ν_π up to a constant.

To begin, note that every atomic refinement $\eta \in R_\pi$ can be associated with a unique triple (A, d, x) where $A \in \pi \setminus \eta$ and

$$\{y \in A \,:\, y_d \leq x\}, \{y \in A \,:\, y_d > x\} \in \eta \setminus \pi. \tag{133}$$

We call the triple (A, d, x) a **guillotine cut** (or simply **cut**) and denote by $\mathscr{C}_\pi$ the set of cuts associated with all atomic refinements R_π, and denote by cut_π the isomorphism mapping $\mathscr{C}_\pi$ onto R_π.

Consider the ring (of subsets of $\mathscr{C}_\pi$) containing the null set as well as sets of the form

$$\{(A, d, x) \,:\, x \in I\}, \tag{134}$$

where $A = I_1 \times \cdots \times I_D \in \pi$, $d \in \{1, \ldots, D\}$, and $I \subseteq I_d$ is a left-open/right-closed subinterval, and consider the real-valued set function $\tilde{\lambda}_\pi$ defined on this ring and given by $\tilde{\lambda}_\pi \emptyset := 0$ and

$$\tilde{\lambda}_\pi \{(A, d, x) \,:\, x \in I\} := |I|. \tag{135}$$

It is straightforward to show that $\tilde{\lambda}_\pi$ is a σ-finite pre-measure and therefore, by Caratheodory's extension theorem, that there exists a

unique extension λ_π to the σ-algebra generated by the ring. Let

$$\nu_\pi := \lambda_\pi \circ \mathsf{cut}_\pi^{-1} \tag{136}$$

denote the induced measure on R_π.

Lemma V.4. *A family $\{\zeta_\pi : \pi \in \mathscr{G}^0\}$ of measures on atomic refinements is invariant under translation, permutation, and restriction if and only if there exists a constant $c \geq 0$ such that for all $\pi \in \mathscr{G}^0$, we have $\zeta_\pi = c \cdot \nu_\pi$.*

Proof. It is straightforward to show that, for all $c \geq 0$, a family satisfying $\zeta_\pi = c \cdot \nu_\pi$ for all $\pi \in \mathscr{G}^0$ is invariant under translation, permutation, and restriction.

In the other direction, assume that a family $\{\zeta_\pi : \pi \in \mathscr{G}^0\}$ is invariant under translation, permutation, and restriction. Let $X \subseteq \mathbb{R}^D$ be a box, let $\pi \in \mathscr{G}^0_X$, and let

$$D_1 = \mathsf{cut}_\pi\text{`}\{(A_1, d_1, x) : x \in I_1\} \quad \text{and,} \tag{137}$$

$$D_2 = \mathsf{cut}_\pi\text{`}\{(A_2, d_2, x) : x \in I_2\} \tag{138}$$

be any two subsets of R_π such that $|I_1| = |I_2|$. We will show that

$$\zeta_\pi D_1 = \zeta_\pi D_2, \tag{139}$$

and thus by the additivity of measures, there exists a constant $c_\pi \geq 0$ such that $\zeta_\pi = c_\pi \cdot \nu_\pi$. Moreover, by invariance under restriction, it follows that $c_\pi = c_\eta$ for all $\pi, \eta \in \mathscr{G}^0$. Therefore, the result follows assuming we can show (139).

Let ρ be a permutation that takes d_2 to d_1 and let τ be a translation such that

$$D_2' := \tau \text{“`}(\rho \text{“`} D_2) = \mathsf{cut}_{\tau \text{“`}(\rho \text{“`} \pi)}\text{`}\{(A_2', d_1, x) : x \in I_1\}, \tag{140}$$

and

$$A_2' \cap A_1 \neq \emptyset, \tag{141}$$

where $A_2' := \tau\text{`}(\rho\text{`}A_2)$. Informally, after the permutation ρ, the translation τ takes I_2 to I_1 along dimension d_1 and enforces some overlap in all other dimensions. By invariance under translation and permutation, we have that

$$\zeta_\pi D_2 = \zeta_{\tau \text{“`}(\rho \text{“`} \pi)} D_2'. \tag{142}$$

Let $Y = A_2' \cap A_1$. Then

$$\Pi_Y \pi = \{Y\} = \Pi_Y(\tau \text{“`}(\rho \text{“`} \pi)) \tag{143}$$

and

$$\Pi_Y\text{`}D_2' = \Pi_Y\text{`}D_1. \tag{144}$$

Therefore,

$$\begin{aligned}
\zeta_\pi D_1 &= \zeta_{\Pi_Y \pi} \Pi_Y \text{`} D_1 && \textit{restriction} && (145)\\
&= \zeta_{\Pi_Y \pi} \Pi_Y \text{`} D_2' && \textit{Eq. (144)} && (146)\\
&= \zeta_{\Pi_Y (\tau \text{“} (\rho \text{“} \pi))} \Pi_Y \text{`} D_2' && \textit{Eq. (143)} && (147)\\
&= \zeta_{\tau \text{“} (\rho \text{“} \pi)} D_2' && \textit{restriction} && (148)\\
&= \zeta_\pi D_2, && \textit{Eq. (142)} && (149)
\end{aligned}$$

completing the proof. □

Let X be a bounded box. By a **uniformly random guillotine cut of** π we mean a random element in $\mathscr{C}_\pi$ with distribution

$$\bar{\lambda}_\pi := \lambda_\pi / \lambda_\pi(\mathscr{C}_\pi), \tag{150}$$

and by a **uniformly random refinement** we mean a random element in R_π with distribution

$$\bar{\nu}_\pi := \nu_\pi / \nu_\pi(R_\pi) = \bar{\lambda}_\pi \circ \mathsf{cut}_\pi^{-1}. \tag{151}$$

(Note that $\lambda_\pi(\mathscr{C}_\pi) < \infty$ if and only if X is a bounded box.)

It follows trivially from (135) that a uniformly random refinement of π splits a subbox $A \in \pi$ with probability proportional to $\sum_d |I_d|$, where $A = I_1 \times \cdots \times I_D$, and, conditioned on splitting $A \in \pi$, cuts the d'th dimension with probability proportional to $|I_d|$.

3.2. Mondrian process. We now construct a Markov process in continuous time that takes values in the space of guillotine partitions. The name was suggested by Matthias Seeger, who thought that the partitions produced by the process resembled the grid-based artwork of the painter Piet Mondrian.

By a **pure jump-type Markov process on a measurable space** $(S, \mathcal{S})$ we mean a Markov process M defined on the index set $\mathbb{R}_+$ that takes values in S and is a.s. right continuous and constant apart from isolated jumps. We will think of M as a random element in the function space $S^{\mathbb{R}_+}$ and write $\mathsf{M}_t = \mathsf{M}(t)$. The distribution of M is completely determined by 1) the distribution of its initial state M_0 and 2) its so-called **rate kernel** $\alpha : S \times \mathcal{S} \to [0, 1]$, which satisfies

$$\alpha(x, B) = \lim_{h \downarrow 0} h^{-1} \mathbf{P}\{\mathsf{M}_{t+h} \in B \mid \mathsf{M}_t = x\} \tag{152}$$

for all $t > 0$, all $x \in S$ and all measurable subsets $B \subseteq S$ such that $x \notin B$. (See [Kal97, Thm. 10.24] for a discussion of the backward equations of pure jump-type Markov processes.)

We begin by defining a Mondrian process on a bounded box:

Definition V.5 (Finite Mondrian process)**.** Let X be a bounded box and let M be a pure jump-type Markov process on $\mathscr{G}_X^0$ with rate kernel α. Then M is a **finite Mondrian process on** X when $\mathsf{M}_0 = \{X\}$ a.s. and

$$\alpha(\pi, B) := \nu_\pi(B \cap R_\pi) \tag{153}$$

for all $\pi \in \mathscr{G}_X^0$ and measurable subsets $B \subseteq \mathscr{G}_X^0$ such that $\pi \notin B$.

We now proceed to the question of the existence of such a process. For $\pi \in \mathscr{G}_X^0$, let $c(\pi) := \nu_\pi(R_\pi)$ denote the **rate function**. Then the rate kernel of a Mondrian process can be written

$$\alpha(\pi, B) = c(\pi)\, \bar{\nu}_\pi(B \cap R_\pi). \tag{154}$$

We will write $\bar{\alpha}$ to denote the **transition kernel** given by $\bar{\alpha}(\pi, B) := \bar{\nu}_\pi(B \cap R_\pi)$.

It is well known that a pure jump-type Markov process can be expressed as a discrete-time Markov chain embedded in continuous time. In particular, let Y be a $\mathscr{G}_X^0$-valued Markov chain in discrete time with initial state $\{X\}$ and transition kernel $\bar{\alpha}$, and let (γ_n) be an i.i.d. sequence of exponentially distributed random variables with mean 1. Define

$$\mathsf{N}_t := Y_n \qquad \text{for } t \in [\tau_n, \tau_{n+1}), \tag{155}$$

where

$$\tau_n := \sum_{j=1}^{n} \frac{\gamma_j}{c(Y_{j-1})}. \tag{156}$$

Note that N_t is well-defined if and only if

$$\tau_\infty := \lim_{n\to\infty} t_n = \infty \quad \text{a.s.} \tag{157}$$

The event $\{\tau_\infty < \infty\}$ is called *explosion*. From (156), we see that $\tau_\infty = \infty$ a.s. if and only if

$$\sum_n \{c(Y_n)\}^{-1} = \infty \quad \text{a.s.} \tag{158}$$

It follows from elementary results on pure jump-type Markov processes (see [Kal97, Thm. 10.19]), that N is a Mondrian process on X if and only if explosion occurs with probability 0.

The possibility of explosion is not obviously ruled out: Note that

$$c(\pi) = \sum_{A\in\pi} c(\{A\}) \tag{159}$$

where

$$c(\{A\}) = \sum_{d=1}^{D} |A_d| \tag{160}$$

for $A = A_1 \times \cdots \times A_D \in \pi$. (Here $|A_d|$ is the length of the interval corresponding to an edge of the box A, and so we will refer to the quantity (160) as the *linear dimension of the box* A.) It follows that $c(\eta) > c(\pi)$ if $\eta \succ \pi$. That is, the rate $c(Y_n)$ almost surely increases. Moreover,

$$\sup\{c(\pi) : \pi \in \mathscr{G}_X^0\} = \infty, \tag{161}$$

and so the rate is unbounded. However, one can show that the rate does not increase too quickly. (See Bertoin [Ber06, Lem. 1.2] for a similar argument.) We begin with a technical lemma:

Lemma V.6 (Bounded growth)**.** *Fix a bounded box X. There is a constant $a > 0$ such that, for all $\pi \in \mathscr{G}_X^0$,*

$$\int_{R_\pi} \bar{\nu}_\pi(d\eta)\, c(\eta) < c(\pi) + a. \tag{162}$$

Proof. Let $\pi \in \mathscr{G}_X^0$. By its construction, ν_π concentrates on the atomic refinements R_π of π. An atomic refinement of π results in a box in π being split in two along some dimension d. The corresponding increase in the total linear dimension equals the additional length of a duplicate edge in all dimensions other than the dth. Every box in π is contained in X, therefore, for $\eta \succ_1 \pi$,

$$c(\eta) < c(\pi) + c(\{X\}). \tag{163}$$

It follows that $a = c(\{X\})$ suffices. □

Theorem V.7 (Existence of a finite Mondrian process)**.** *Let X be a bounded box. Then there exists a Mondrian process on X.*

Proof. Let Y and (τ_n) be the embedded Markov chain and jump times, respectively, as defined above. It suffices to show that Condition (158) holds.

By the law of iterated expectations and Lemma V.6, we have

$$\mathbb{E}\{c(Y_n)\} = \mathbb{E}\{\mathbb{E}[c(Y_n) \mid Y_{n-1}]\} \tag{164}$$

$$< \mathbb{E}\{c(Y_{n-1})\} + a. \tag{165}$$

Therefore, $\mathbb{E}\{c(Y_n)\} = \mathbb{E}\{c(Y_0)\} + an$. It follows from Fatou's lemma that

$$\mathbb{E} \liminf_{n\to\infty} \frac{c(Y_n)}{n} \leq \liminf_{n\to\infty} \mathbb{E}\frac{c(Y_n)}{n} \tag{166}$$

$$< \liminf_{n\to\infty} \frac{\mathbb{E}\{c(Y_0)\} + an}{n} \tag{167}$$

$$< \infty, \tag{168}$$

and so

$$\liminf_{n\to\infty} \frac{c(Y_n)}{n} < \infty \quad \text{a.s.} \tag{169}$$

Because $c(Y_n)$ is almost surely increasing, $\sum_{n=1}^{\infty} \{c(Y_n)\}^{-1} = \infty$ a.s., completing the proof. □

Let M be a Mondrian process in one-dimension on an interval $X \subseteq \mathbb{R}$. Then for all $t \geq 0$, M_t is an a.s. finite partition of the interval that we can represent by the set of points at the boundaries of the interval partitions. Each atomic refinement introduces an additional interval boundary point, and so we can therefore think of a Mondrian process in one-dimension as a (pure jump-type Markov) point process. There is a close connection between the Mondrian process and the Poisson process, the canonical point process.

Theorem V.8. *Let M be a Mondrian process in one-dimension on an interval X. Then the point process associated with the boundaries of the interval partitions is a homogeneous Poisson point process on X.*

Proof. Note that for any partition $\pi \in \mathscr{G}_X^0$ of the interval X, the rate satisfies $c(\pi) = |X|$. Therefore, the wait time between jumps are independent and exponentially distributed with rate $|X|$. Moreover, $\bar{\nu}_\pi$ is equal in distribution to the uniform distribution on X, and so conditioned on the number of cuts, they are uniformly distributed on X. It is an elementary fact that these two properties imply that the boundaries are a Poisson point process (see [Kal97, Cor. 10.10]). □

Remark V.9. Consider a Mondrian process M on the unit interval and, for each time $t \geq 0$, let $L_1(t) \geq L_2(t) \geq \cdots$ be the sorted lengths of the intervals comprising the partition M_t, where $L_k(t) = 0$ when there are fewer than k intervals at time t. The sequence-valued process $L = (L_1, L_2, \dots)$ is an example of a *fragmentation chain* (see [Ber06, §1.1.3 and §2.1.2] for more details on this "Poissonian rain" process).

A fragmentation chain is a Markov process in continuous time whose state is a (possibly countably infinite) population of particles, each entirely characterized by its *mass*, a nonnegative real quantity.

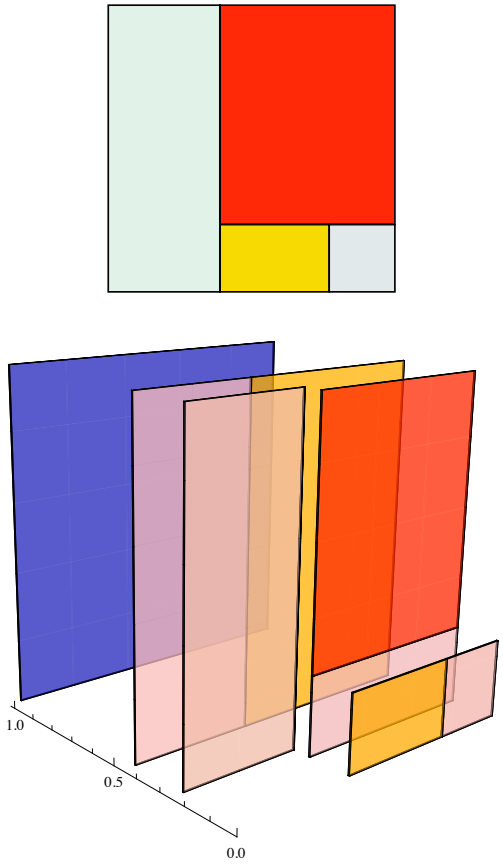

FIGURE 4. (top) Mondrian process on the unit square at time $t = 1$. (bottom) A visualization of the same Mondrian process for $t \in [0, 1]$. Note: the colors are chosen at random.

Fragmentation chains are one example of a more general class of *fragmentation processes*, which have been fruitfully applied in various areas of physics, chemistry, biology, statistics and computer science (see [Ber06] for an introduction).

The particles in a fragmentation chain evolve independently, randomly fragmenting into subpopulations of particles of smaller mass. In the Poissonian rain process describe above, the subintervals can be considered to be particles in a fragmentation chain and their lengths play the role of their mass.

Bertoin [Ber06] presents various generalizations of Poissonian rain that are also fragmentation chains, including higher-dimensional analogues. We will show that Mondrian processes can also be viewed as a non-interacting particle systems, in the sense that each box (particle)

evolves independently of the other boxes. One might wonder whether the volume of each box could play the role of a particle's mass in a fragmentation chain. However, the evolution of the ranked volumes of a Mondrian process is not Markovian. On the other hand, a Mondrian processes in D dimensions can be thought of as a multi-dimensional generalizations of a fragmentation process, where a one-dimensional mass is replaced by a D-dimensional quantity, in this case representing the length of the box along each dimension. We believe that multi-dimensional generalizations of fragmentation processes are likely to be useful for constructing other kinds of random data structures.

We now proceed to characterize the structure of Mondrian processes:

Henceforth, we will consider a Mondrian process M defined on an underlying probability space. For all $\pi \in \mathscr{G}^0$, we will denote by $\mathbf{P}_\pi$ the probability measure for which $\mathsf{M}_0 = \pi$ a.s. (i.e., under $\mathbf{P}_\pi$, the Mondrian process starts from π rather than the trivial partition).

Proposition V.10 (Recursive construction)**.** *Let $\pi \in \mathscr{G}_X^0$ and let $\{\mathsf{N}^A\}_{A\in\pi}$ be independent Mondrian processes on $A \in \pi$. Then the $\mathscr{G}_X^0$-valued stochastic process M on $\mathbb{R}_+$ given by*

$$\mathsf{M}_t := \bigcup_{A\in\pi} \mathsf{N}_t^A \tag{170}$$

is a Mondrian process under $\mathbf{P}_\pi$.

Before we proceed with the proof, we state two important properties of exponential random variables upon which we will rely:

Lemma V.11 (Memoryless)**.** *Let ζ be an exponential random variable. Then for $s, t \geq 0$, we have that $\mathbf{P}\{\zeta \geq t \mid \zeta \geq s\} = \mathbf{P}\{\zeta \geq t - s\}$.* □

Lemma V.12 (Poisson splitting)**.** *Fix a positive integer k and let $(\zeta_n : n \leq k)$ be independent exponentially distributed random variables with rates $r_n > 0$, respectively. Then $\zeta = \min_n \zeta_n$ is exponentially distributed with rate $\sum_n r_n$. Moreover, the events $\{\zeta = \zeta_n\}$ are independent of ζ, a.s. disjoint, and occur with probability $\mathbf{P}\{\zeta = \zeta_n\} = \frac{r_n}{\sum_n r_n}$.* □

Proof of Proposition V.10. Note that $\Pi_A \circ \mathsf{M} = \mathsf{N}^A$ by construction. Let Y^A and $\tau_1^A < \tau_2^A < \cdots$ be the embedded Markov chain and jump times of N^A, respectively. For $t \geq 0$, define n_t^A to be the (a.s. unique) index such that $t \in [\tau_{n_t^A}^A, \tau_{n_t^A+1}^A)$. Then

$$\mathsf{M}_t = \bigcup_{A\in\pi} Y_{n_t^A}^A. \tag{171}$$

It is clear from the independence of the chains and jump times that M is a.s. right continuous and constant apart from isolated jumps.

For $t \geq 0$, define

$$\zeta_t^A := \tau^A_{n_t^A+1} - t \tag{172}$$

to be the wait time until the next jump to $\mathsf{N}^A = \Pi_A \circ \mathsf{M}$ after time t Then the wait time until the next jump to M after time t is

$$\zeta_t = \min_{A \in \pi} \zeta_t^A. \tag{173}$$

Conditioned on $\mathcal{F}_t := \sigma\{\mathsf{M}_s : s \leq t\}$, it follows from the memoryless property of exponential random variables that ζ_t^A is exponentially distributed with rate $c(Y^A_{n_t^A}) = c\{\Pi_A(\mathsf{M}_t)\}$. By the mutual independence of $\{(Y^A, \tau^A)\}_{A \in \pi}$, it follows that ζ_t is $\mathcal{F}_t$-conditionally exponentially distributed with rate

$$\sum_{A \in \pi} c\{\Pi_A(\mathsf{M}_t)\} = c(\mathsf{M}_t). \tag{174}$$

Moreover, conditioned on $\mathcal{F}_t$, the (a.s. unique) set $\alpha_t \in \pi$ that achieves the minimum in (173) is independent of ζ_t and equal to $A \in \pi$ with probability

$$\frac{c\{\Pi_A(\mathsf{M}_t)\}}{c\{\mathsf{M}_t)}. \tag{175}$$

Conditioning on $\{\alpha_t = A\}$ and $\mathcal{F}_t$, we have that

$$\Pi_A(\mathsf{M}_{t+\zeta_t}) = Y^A_{n_t^A+1} \tag{176}$$

is independent of ζ_t and thus has distribution

$$\bar{\nu}_{Y^A_{n_t^A}} = \bar{\nu}_{\Pi_A \mathsf{M}_t}. \tag{177}$$

Therefore, conditioned on $\mathcal{F}_t$, $\mathsf{M}_{t+\zeta_t}$ is independent of ζ_t and has distribution

$$\bar{\nu}_{\mathsf{M}_t}. \tag{178}$$

Along with the trivial observation that $\mathsf{M}_0 = \pi$ a.s., it follows that M is a pure jump-type Markov process with initial state π and rate kernel given by (153), completing the proof. □

The previous proof shows that we can construct a Mondrian process from independent Mondrian processes defined on the boxes of a partition. We can use this generative construction to gain representational insight into Mondrian processes using a so-called transfer argument (see [Kal97, Thm. 5.10] for more details).

Theorem V.13 (Transfer)**.** *Fix any measurable space S and Borel space T, and let $\xi \stackrel{d}{=} \tilde{\xi}$ and η be random elements in S and T, respectively. Then there exists a random element $\tilde{\eta}$ in T with $(\tilde{\xi}, \tilde{\eta}) \stackrel{d}{=} (\xi, \eta)$.* □

The next result shows that, in fact, all Mondrian processes are composed from independent Mondrian processes.

Theorem V.14 (Branching property)**.** *Let $\pi \in \mathscr{G}_X^0$, and let M be a Mondrian process on X under $\mathbf{P}_\pi$. Then $\{\Pi_A \circ \mathsf{M}\}_{A\in\pi}$ are independent Mondrian processes.*

Proof. Let $\{\hat{\mathsf{N}}^A\}_{A\in\pi}$ be independent Mondrian processes on $A \in \pi$, respectively. By Proposition V.10, there is a Mondrian process $\hat{\mathsf{M}}$ on X, and in particular $\hat{\mathsf{M}} \stackrel{d}{=} \mathsf{M}$. Hence, by Theorem V.13, there are processes N^A for each $A \in \pi$ such that

$$(\hat{\mathsf{M}}, \hat{\mathsf{N}}^A \,:\, A \in \pi) \stackrel{d}{=} (\mathsf{M}, \mathsf{N}^A \,:\, A \in \pi). \tag{179}$$

As $\{\mathsf{N}^A\}$ are independent Mondrian processes and $\Pi_A\hat{\mathsf{M}}_t = \hat{\mathsf{N}}_t^A$ a.s. for all $t \geq 0$ and $A \in \pi$, the same properties and relationships hold of M and $\{\mathsf{N}^A\}$, completing the proof. □

Remark V.15. It is tempting to consider a direct proof. We present an informal but direct argument in the proof of Theorem V.24.

Let M be a Mondrian process on a bounded box X, and define $\theta_t(s) = s + t$. Then $\mathsf{M} \circ \theta_t$ is a Mondrian process shifted ahead in time by an amount t, i.e., a Mondrian process under $\mathbf{P}_{\mathsf{M}_t}$. This property also holds for an optional time τ. In particular, by the strong Markov property, conditioned on the process up until τ, the evolution of $\mathsf{M} \circ \theta_\tau$ is given by $\mathbf{P}_{\mathsf{M}_\tau}$. Therefore, conditioning on M_τ, it follows from Theorem V.14 that the time-shifted restrictions $\{\Pi_A \circ \mathsf{M} \circ \theta_\tau\}_{A\in\mathsf{M}_\tau}$ are conditionally independent Mondrian processes.

In particular, taking the optional time τ to be the first jump time of M, this observation suggests the follow recursive construction of a Mondrian process: Let ξ be an exponential random variable with rate $c(\{X\})$, let $\{X_0, X_1\}$ be a uniformly random refinement of $\{X\}$, and let M^0 and M^1 be independent Mondrian processes on X_0 and X_1, respectively. Then the process whose first jump is the atomic refinement $\{X_0, X_1\}$ at time ξ, and subsequent evolution is given by $t \mapsto \bigcup_{i\in 2} \mathsf{M}^i_{t-\xi}$ is itself a Mondrian process. (Here 2 stands for the two point set $\{0, 1\}$.) Moreover, the independent Mondrian processes can themselves be generated recursively. This leads to the following algorithm for sampling M_t on a box $X \subseteq \mathbb{R}^D$:

SAMPLEM(t, X)
1. $\xi \sim \text{Exponential}(c(\{X\}))$
2. **if** $t < \xi$
3. **return** trivial partition $\{X\}$
4. **otherwise**
5. Sample an atomic refinement $\{X_0, X_1\} \sim \bar{\nu}_{\{X\}}$
6. **return** $\bigcup_{i\in 2}$ SAMPLEM$(t - \xi, X_i)$.

This recursive procedure highlights one of the computational implications of conditional independence, and in particular, the branching property: the future evolution of one part of the partition depends only on a local structure of the partition. In the next section we study conditional distributions of Mondrian processes. We use these results to show that a Mondrian process is *self-similar* under restriction, i.e., the restriction of a Mondrian process to an arbitrary subbox is itself a Mondrian process. This property will allow us to construct a Mondrian process on an infinite space. Again, the branching property is useful because it allows us to extend a Mondrian process locally.

4. Conditional Mondrian processes

Let $A \subseteq X$ be bounded boxes and let M be a Mondrian process on X. We are interested in characterizing the conditional distribution of M given its restriction $\Pi_A \circ \mathsf{M}$ to A. We will begin by describing a way to extend a Mondrian process to a larger space.

Let N be a Mondrian process on A, and let Z and $\sigma_1 < \sigma_2 < \cdots$ be the embedded Markov chain and jump times of N, respectively. For $t \geq 0$, define m_t to be the (a.s. unique) index such that $t \in [\sigma_{m_t}, \sigma_{m_t+1})$. That is, for all $t \geq 0$, $\mathsf{N}_t = Z_{m_t}$.

Given an atomic refinement η to $\Pi_A\pi$, there is a unique atomic refinement to π that agrees with η by restriction. Formally, for all $\pi \in \mathscr{G}_X^0$ and $\eta \succ_1 \Pi_A(\pi)$, let $\mathit{lift}_{\pi,A}(\eta)$ denote the unique atomic refinement to π such that $\Pi_A(\mathit{lift}_{\pi,A}(\eta)) = \eta$.

Finally, for $\pi \in \mathscr{G}_X^0$ and a uniformly distributed random variable θ in $[0, 1]$, let $\mathit{gencut}_A(\pi, \theta)$ be a random refinement of π equal in distribution to that of a uniformly random refinement η conditioned on the event $\{\Pi_A(\eta) = \Pi_A(\pi)\}$. Put simply, $\mathit{gencut}_A(\pi, \theta)$ is a uniformly random refinement of π whose associated cut does not cross the subbox A.

We now define an embedded Markov chain on $\mathscr{G}_X^0$ and jump times. Put $\tau_0 := 0$ and $Y_0 := \{X\}$ and for $n \in \mathbb{N}$, define τ_{n+1} and Y_{n+1}

inductively by

$$\tau_{n+1} := \min\{\sigma_{m_{\tau_n}+1},\ \tau_n + \frac{\xi_n}{c(Y_n) - c(Z_{m_{\tau_n}})}\}, \tag{180}$$

and

$$Y_{n+1} := \begin{cases} \mathit{lift}_{Y_n,A}(Z_{m_{\tau_n}+1}) & \text{if } \tau_{n+1} = \sigma_{m_{\tau_n}+1} \\ \mathit{gencut}_A(Y_n, \theta_n) & \text{otherwise,} \end{cases} \tag{181}$$

where $(\xi_n : n \in \mathbb{N})$ is an independent sequence of i.i.d. exponential random variables with mean 1, and $(\theta_n : n \in \mathbb{N})$ is an independent sequence of i.i.d. $U[0,1]$ random variables.

Proposition V.16 (Well-definedness)**.** *The processes* $(Y_n : n \in \mathbb{N})$ *and* $(\tau_n : n \in \mathbb{N})$ *are well-defined.*

Proof. A sufficient condition for these equations to be well-defined is that

$$Z_{m_{\tau_n}+1} \succ_1 \Pi_A(Y_n), \qquad \forall n \in \mathbb{N}. \tag{182}$$

First note that $Z_{m_{\tau_n}+1} \succ_1 Z_{m_{\tau_n}}$ holds for all $n \in \mathbb{N}$. In order to establish (182), we will show that

$$Z_{m_{\tau_n}} = \Pi_A(Y_n) \tag{183}$$

holds for all $n \in \mathbb{N}$. To begin, observe that

$$\Pi_A(Y_0) = \{A\} = Z_0 = Z_{m_{\tau_0}}. \tag{184}$$

Assume that (183) holds for all $m \leq n \in \mathbb{N}$. If $\tau_{n+1} \neq \sigma_{m_{\tau_n}+1}$, then $m_{\tau_{n+1}} = m_{\tau_n}$ and, by (181) and the inductive hypothesis,

$$\Pi_A(Y_{n+1}) = \Pi_A(Y_n) = Z_{m_{\tau_n}} = Z_{m_{\tau_{n+1}}}, \tag{185}$$

as desired. Now assume that $\tau_{n+1} = \sigma_{m_{\tau_n}+1}$. By the inductive hypothesis,

$$Z_{m_{\tau_n}+1} \succ_1 Z_{m_{\tau_n}} = \Pi_A(Y_n) \tag{186}$$

and so, by (181) and the definition of *lift*, we have

$$\Pi_A(Y_{n+1}) = \Pi_A(\mathit{lift}_{Y_n,A}(Z_{m_{\tau_n}+1})) = Z_{m_{\tau_n}+1}. \tag{187}$$

Yet $m_{\tau_{n+1}} = m_{\tau_n} + 1$ and so

$$\Pi_A(Y_{n+1}) = Z_{m_{\tau_{n+1}}}, \tag{188}$$

completing the proof. □

For $t \geq 0$, define n_t to be the (a.s. unique) index such that $t \in [\tau_{n_t}, \tau_{n_t+1})$, and consider the $\mathscr{G}_X^0$-valued stochastic process $\hat{\mathsf{M}}$ on $\mathbb{R}_+$ given by

$$\hat{\mathsf{M}}_t := Y_{n_t}. \tag{189}$$

Theorem V.17 (Conditional Mondrian process)**.** *Let $A \subseteq X$ be bounded boxes and let N be a Mondrian process on A. Then a process $\hat{\mathsf{M}}$ as defined by (189) is a Mondrian process on X and $\Pi_A \circ \hat{\mathsf{M}} = \mathsf{N}$ a.s.*

Proof. We begin by establishing the well-definedness of n_t, which requires that $\tau_n \to \infty$ a.s. One can see from (181) that $Y_{n+1} \succ_1 Y_n$, and so the chain Y is a sequence of atomic refinements starting with $Y_0 = \{X\}$ and thus an element in $\mathscr{G}_X^0$. Therefore, by Lemma V.6, it follows that $c(Y_n) \leq c(Y_0) + an$ a.s. for some constant $a > 0$, and so

$$\sum_n \frac{1}{c(Y_n) - c(Z_{m_{\tau_n}})} \geq \sum_n \frac{1}{c(Y_n)} \geq \sum_n \frac{1}{c(Y_0) + an} = \infty. \tag{190}$$

Combined with the independence of (ξ_n) and the fact that $\sigma_n \to \infty$, it follows that $\tau_n \to \infty$ a.s. We may then also conclude that $\hat{\mathsf{M}}$ is a.s. right continuous and constant apart from isolated jumps.

Let $t \geq 0$. Then the wait time until the next jump in $\hat{\mathsf{M}}$ is

$$\zeta_t := \tau_{n_t+1} - t. \tag{191}$$

Conditioned on $\mathcal{F}_t = \sigma\{\hat{\mathsf{M}}_s : s \leq t\}$, ζ_t is exponentially distributed with rate $c(Z_{m_{\tau_{n_t}}}) + c(Y_{n_t}) - c(Z_{m_{\tau_{n_t}}}) = c(Y_{n_t}) = c(\hat{\mathsf{M}}_t)$. Moreover, the event

$$\{\tau_{n_t+1} = \sigma_{m_{\tau_{n_t}}+1}\} \tag{192}$$

is independent of ζ_t and occurs with probability

$$\frac{c(Z_{m_{\tau_{n_t}}})}{c(\hat{\mathsf{M}}_t)} = \frac{c(\mathsf{N}_t)}{c(\hat{\mathsf{M}}_t)}. \tag{193}$$

Moreover, conditioned on $\mathcal{F}$ and (192), the restriction of the refinement is a uniformly random refinement; likewise, conditioned on $\mathcal{F}$ and the complement of (192), the refinement is uniformly random outside of A. Taking expectations conditioned on $\mathcal{F}$, the distribution of the refinement is $\bar{\nu}_{\hat{\mathsf{M}}_t}$, and this distribution is independent of ζ_t. Finally, we note that $\hat{\mathsf{M}}_0 = \{X\}$ a.s.

Therefore, $\hat{\mathsf{M}}$ is a Mondrian process on X. Moreover, by construction $\Pi_A(\hat{\mathsf{M}}_t) = \Pi_A(Y_{n_t}) = Z_{m_t} = \mathsf{N}_t$ a.s. □

Remark V.18. Informally, the next jump to M comes from a race between 1) jumps inside A, i.e., jumps in N and 2) jumps outside A. As both are independent, exponential random variables, their minimum is exponentially distributed with a rate given by the sum of the rates of jumps coming from both sources. When a jump occurs in N, there is a unique cut to the current refinement which achieves that cut when restricted to A. Otherwise, a uniformly random cut is made outside A.

Let $A \subseteq X \subseteq \mathbb{R}^D$ be boxes, and let N be a Mondrian process on A. As before, the branching property (Theorem V.14) of the Mondrian process allows us to give a recursive construction of a Mondrian process M on X extending N, i.e., satisfying $\Pi_A \circ \mathsf{N} = \mathsf{M}$. This leads to the following recursive algorithm for sampling M_t given N:

Let t_1 and Y_1 be the first jump time and transition of N, and recall that $\mathsf{N}' := \mathsf{N} \circ \theta_s$ is a time-shifted version of N, satisfying $\mathsf{N}'_t = \mathsf{N}_{s+t}$.

SampleCM$(t,\ X \mid \mathsf{N},\ A)$,
1. $\xi \sim \text{Exponential}(c(\{X\}) - c(\{A\}))$
2. **if** $\min\{t, t_1\} < \xi$
3. **if** $t < t_1$
4. **return** trivial partition $\{X\}$
5. **otherwise**
6. $\{X_0, X_1\} := \mathit{lift}_{Y_1,A}(\{X\})$
7. **otherwise**
8. $\{X_0, X_1\} \sim \mathbf{P}\{\mathit{gencut}_A(\{X\}, \theta) \in \cdot\}$
9. **return** $\bigcup_{i\in 2}$ SampleCM$(t - \xi,\ X_i \mid \Pi_{X_i} \circ \mathsf{N} \circ \theta_\xi,\ A \cap X_i)$

As with Proposition V.10 and Theorem V.14, this generative result can give us representational insight via the use of a transfer argument. In particular, recall that guillotine partitions are closed under restriction, and that our notion of a uniformly random cut was based on the desired property of invariance under restriction. Mondrian processes enjoy a similar closure property under restriction:

Theorem V.19 (Self-similarity under restriction)**.** *Let $A \subseteq X$ be bounded boxes and let M be a Mondrian process on X. Then $\Pi_A \circ \mathsf{M}$ is a Mondrian process on A.*

Proof. Let $\hat{\mathsf{N}}$ be Mondrian process on A. By Theorem V.17, there exists a Mondrian process $\hat{\mathsf{M}}$ on X such that $\Pi_A \circ \hat{\mathsf{M}} = \hat{\mathsf{N}}$ a.s. As $\hat{\mathsf{M}} \stackrel{d}{=} \mathsf{M}$, it follows from Theorem V.13, that there exists a process N' such that $(\hat{\mathsf{M}}, \hat{\mathsf{N}}) \stackrel{d}{=} (\mathsf{M}, \mathsf{N}')$. In particular, $\hat{\mathsf{N}} \stackrel{d}{=} \mathsf{N}'$ and $\mathsf{N}' = \Pi_A \circ \mathsf{M}$ a.s., and so $\Pi_A \circ \mathsf{M}$ is a Mondrian process on A. □

Remark V.20. Theorem V.17 and Theorem V.19 give us a complete characterization of the conditional distribution of a Mondrian process given a restriction of that process.

Recall that the class of guillotine partitions are closed under restriction. Here we see that Mondrian processes enjoy a similar property. It is straightforward to show that the Mondrian process is also self-similar under translation and permutation.

Heretofore, representational results were derived from generative results. Another approach is to characterize directly the representation. The following are informal arguments that give additional insight into the structure of Mondrian processes.

Definition V.21. Let X be a box, let $\mathcal{A} \subseteq \mathcal{P}(X)$, and let $\pi \in \mathscr{G}_X^0$. We say that $\mathcal{A}$ is **π-separated** when $\mathcal{A}$ is a collection of boxes in X such that $A \subseteq B \in \pi$ for all $A \in \mathcal{A}$ and, for all $\eta \succ_1 \pi$, there is at most one element $A \in \mathcal{A}$ such that $\eta|_A \neq \pi|_A$.

The following results are immediate from the definition of an atomic refinement:

Proposition V.22. *Let $\pi \in \mathscr{G}_X^0$. Then π is π-separated.* □

Proposition V.23. *Let $A, B \subseteq X \subseteq \mathbb{R}^D$ be boxes. Then $\{A, B\}$ is $\{X\}$-separated if and only if $A_d \cap B_d = \emptyset$ for all $d \in \{1, \ldots, D\}$.* □

Theorem V.24 (Independence of separated restrictions)**.** *Let $\pi \in \mathscr{G}_X^0$ and let $\mathcal{A} \subseteq \mathcal{P}(X)$ be π-separated. Under $\mathbf{P}_\pi$, $\{\Pi_A \circ \mathsf{M}\}_{A \in \mathcal{A}}$ are independent Mondrian processes.*

Before we proceed to sketch a proof of Theorem V.24, we will pause to point out two key facts concerning the rate kernel of a Mondrian process: For each partition $\eta \in \mathscr{G}_X^0$ and box $B \subseteq X$, define

$$R_\eta^B := \{\eta' \succ_1 \eta \,:\, \Pi_B(\eta') \neq \Pi_B(\eta)\} \tag{194}$$

to be the set of atomic refinements of η resulting from a cut through B. Note that, by invariance under restriction, the ν_η-measure of cuts to η that cross B is

$$\nu_\eta(R_\eta^B) = \nu_{\Pi_B(\eta)}(R_{\Pi_B(\eta)}) = c(\Pi_B(\eta)), \tag{195}$$

which *does not depend on η other than through its restriction* $\Pi_B(\eta)$. Moreover, if η' is a uniformly random refinement of η, then conditioned on the event $\{\eta' \in R_\eta^B\}$, invariance under restriction also implies that the distribution of $\Pi_B(\eta')$ is $\bar{\nu}_{\Pi_B(\eta)}$.

PROOF SKETCH OF THEOREM V.24. Let $A \in \mathcal{A}$, and define $\mathsf{N}^A := \Pi_A \circ \mathsf{M}$. Then $\mathsf{N}^A_0 = \Pi_A(\pi) = \{A\}$ with $\mathbf{P}_\pi$-probability 1. Moreover, it follows from (195) that, conditioned on the process up until time t, the rate of jumps to N^A is $c(\mathsf{N}^A_t)$. Therefore, conditioned on N^A_t, the wait time until the next jump is exponentially distributed with inverse mean $c(\mathsf{N}^A_t)$. Moreover, when the jump occurs, it is independent of the wait time and distributed according to $\bar{\nu}_{\mathsf{N}^A_t}$. Therefore, N^A is a Mondrian process on A.

Recall that $\mathcal{A}$ is π-separated. It follows that N^A does not interact with any other restriction N^B, for $B \in \mathcal{A} \setminus \{A\}$, because an atomic refinement that crosses N^A leaves N^B unchanged. Therefore, the processes $\{\mathsf{N}^A\}_{A \in \mathcal{A}}$ are independent. □

Remark V.25. The previous theorem is a slight generalization of the transfer results in that the earlier results imply the result in the special case that no two sets $A, A' \in \mathcal{A}$ are subsets of the same box $B \in \pi$. In order to close the gap, it would be necessary to extend the conditional Mondrian construction from a restriction to a π-separated collection of restrictions. We will not pursue this here, as it is a relatively straightforward extension of the presented results.

5. Mondrian processes on unbounded spaces

The fact that any restriction of a finite Mondrian process is itself a finite Mondrian process hints at the interesting possibility that there is a partition-valued Markov process on $\mathbb{R}^D$, every bounded restriction of which is a finite Mondrian process. In fact, we can give an explicit construction of such a process. We begin by defining a suitable space of partitions:

Definition V.26 (σ-finite guillotine partitions)**.** The set $\mathscr{G}_X$ of **σ-finite guillotine partitions** on X is defined to be the subset of floorplan partitions $\pi \in \mathscr{F}_X$ such that, for all bounded boxes $A \subseteq X$, we have $\Pi_A(\pi) \in \mathscr{G}^0_A$.

Remark V.27 (Rootlessness of σ-finite guillotine partitions)**.** Every guillotine partition has at least one root cut as there is at least one finite sequence of cuts that generates the partition. The same is not true of σ-finite guillotine partitions in general. Some may have no root cut, in which case the hierarchy of atomic refinements will be an infinite tree where each restriction of the σ-finite guillotine partition to a bounded box corresponds with a particular finite subtree.

Just as closedness under restriction motivated our definition of σ-finite guillotine partitions, self-similarity under restriction of finite

Mondrian processes motivates the following definition of σ-finite Mondrian processes.

Definition V.28 (σ-finite Mondrian process)**.** Let $X \subseteq \mathbb{R}^D$ be a box. A $\mathscr{F}_X$-valued Markov process M in continuous time is a **σ-finite Mondrian process on** X when, for all bounded boxes $A \subseteq X$, $\Pi_A \circ \mathsf{M}$ is a finite Mondrian process on A.

We now demonstrate the existence of such a process by constructing it explicitly:

Theorem V.29 (Existence of σ-finite Mondrian processes)**.** *Let X be a (possibly unbounded) box. Then there exists a σ-finite Mondrian process on X.*

Proof. Let $X_0 \subseteq X_1 \subseteq X_2 \subseteq \cdots$ be a nested sequence of bounded boxes such that $\bigcup_n X_n = X$, and let M^0 be a finite Mondrian process on X_0. For all $n \in \mathbb{N}$, construct M^{n+1} from M^n as in (189). It is then the case that M^n is a finite Mondrian process on X_n and $\Pi_{X_m} \circ \mathsf{M}^n = \mathsf{M}^m$ a.s. for all $m \leq n \in \mathbb{N}$.

For all $t \geq 0$, define

$$\mathsf{M}_t^\infty := \Big\{ \bigcup_{n \geq n_0} B_n \,:\, \exists n_0 \in \mathbb{N},\ \forall n \geq n_0, \\ B_n \in \mathsf{M}_t^n \text{ and } B_n \subset B_{n+1} \Big\}. \quad (196)$$

It remains to be shown that M_t^∞ is a floorplan partition and that, for all bounded boxes A, $\Pi_A \circ \mathsf{M}^n$ is a finite Mondrian process on A.

We begin by arguing that M_t^∞ is a floorplan partition. For all $n \in \mathbb{N}$ and $t \geq 0$, $B \in \mathsf{M}_t^n$ implies that, for all $m \geq n$, there exists an a.s. unique $B_m \in \mathsf{M}_t^m$ such that $B \subseteq B_m$, and in particular $B \cup B_m = B_m$. Given that $\bigcup_n X_n$ is a box (namely, X), it follows that, with probability one, the elements of M_t^∞ are themselves boxes. Moreover, these boxes are a.s. disjoint. To see this, note that if distinct $B, B' \in \mathsf{M}_t^\infty$ and $B \cap B' \neq \emptyset$, then there exists a finite $n \in \mathbb{N}$ such that there exist distinct $C, C' \in \mathsf{M}_t^n$ with $C \subseteq B$ and $C' \subseteq B'$, but $C \cap C' \neq \emptyset$, a null event given that M_t^n is a guillotine-partition. Finally, M^∞ is exhaustive (and thus a floorplan partition) as every point x is eventually included in some box $B \in \mathsf{M}_t^n$ and thus is included in some box $B' \in \mathsf{M}_t^\infty$ where $B \subseteq B'$.

Let $m \in \mathbb{N}$. We now argue that $\Pi_{X_m}(\mathsf{M}_t^\infty) = \mathsf{M}_t^m$. Let $B \in \mathsf{M}_t^m$. Then $B \subseteq C := \bigcup_{n \geq m} B_n \in \mathsf{M}_t^\infty$ where, for all $n \geq m$, B_n is the unique set in M_t^n such that $B_m \subseteq B_n$. Let $B' = X_n \cap C$. Clearly $B \subseteq B'$. If $D := B' \setminus B \neq \emptyset$ then there exists some $n \geq m$ such that $D \subseteq B_n$. But

$\Pi_{X_n}(\mathsf{M}_t^n) = \mathsf{M}_t^m$ and so $B \cap B_n = B$, a contradiction. It follows that $\Pi_{X_m}(\mathsf{M}_t^\infty) = \mathsf{M}_t^m$.

For any bounded box A, there exists an n such that $A \subseteq X_n$. Then,

$$\Pi_A \circ \mathsf{M}^\infty = \Pi_A \circ \Pi_{X_n} \circ \mathsf{M}^\infty = \Pi_A \circ \mathsf{M}^n. \tag{197}$$

By construction, M^n is a finite Mondrian process on X_n, and so by Theorem V.19, $\Pi_A \circ \mathsf{M}^n$ is a finite Mondrian process on A, completing the proof. □

As a final remark, note that, while a Mondrian process is composed of isolated jumps, this is not the case for a σ-finite Mondrian process on the entirety of $\mathbb{R}^D$; it is a Markov process but not a pure jump-type Markov process. In particular, at every point in time, $c(\mathsf{M}_t) = \infty$ and so, during every interval of time, an infinite number of atomic refinements will occur. Nevertheless, we can sample the restriction of a σ-finite Mondrian process on any bounded box, and then extend the process to a larger, enclosing box as needed.

CHAPTER VI

Conclusion

In this dissertation, we have studied two core concepts in probability theory and Bayesian statistics through the lens of computability theory. In the first case, we characterized the computability of conditional probability by giving examples of computable joint distributions with noncomputable conditional distributions, and by proving the sufficiency of certain types of structure, like computable densities and sufficiently smooth computable noise, for computing conditional distributions. In the second case, we studied exchangeable sequences and their de Finetti measures, showing that one is computable exactly when the other is computable. This result allowed us to give necessary and sufficient conditions for the computability of posterior distributions of directing random measures. Despite the noncomputability of conditional probability in general, these various positive results cover a considerable fraction of the kinds of probabilistic models studied within finite-dimensional and nonparametric Bayesian statistics.

The question of the existence of algorithms, i.e., the question of computability, is fundamental to the development, study and use of probabilistic programming languages within AI and Bayesian statistics. With our negative results on conditional distributions presented in Chapter III, we have demonstrated fundamental limitations on the generality of probabilistic programming inference algorithms that aim to support inductive reasoning. On the other hand, the various positive results that we have shown can be used by language designers to add inferential support for continuous random variables in limited settings.

We have also connected stochastic processes and random data structures and presented a case study where we construct an infinite random kd-tree data structure, motivated by the problem of modeling relational data. Specifically, in Chapter V, we presented a new construction of the Mondrian process as a pure jump-type Markov process in continuous time, allowing us to use many classical results to study its properties. In particular, by employing a "transfer" argument, we proved that Mondrian processes enjoy a "self-similarity" property, which we used to define and construct infinite versions of Mondrian processes. The

structure of these so-called σ-finite Mondrian processes is that of an infinite kd-tree data structures. In addition, we demonstrated a close connection between Poisson processes and Mondrian processes in one dimension, and have argued that the Mondrian process can be seen as one example of a new class of *multidimensional* fragmentation processes. As fragmentation processes underlie many Bayesian nonparametric priors, this connection between Mondrian processes and fragmentation processes suggests that we are likely to find further applications of multidimensional fragmentation processes.

There are many avenues that deserve further investigation. One in particular is the study of the computability and complexity of *partial* exchangeability, which will likely elucidate deep connections between representation and efficiency, and enable us to answer questions such as, which random data structures admit efficient inference? Other computability results that we have presented have analogues in the polynomial-time setting, and there is a rich literature on probabilistic inference in graphical models; average-case complexity and cryptographic hardness; as well as recent work on the complexity of real continuous functions that can serve as a foundation for further study.

This dissertation has explored the boundary of computability, inference and modeling in probabilistic programming. In closing, we believe that an understanding of the inherent limitations and possibilities of a probabilistic programming approach to AI will be increasingly relevant in a world populated by complex systems that must manage their own uncertainty.

Bibliography

[AES00] M. Alvarez-Manilla, A. Edalat, and N. Saheb-Djahromi. An extension result for continuous valuations. *J. London Math. Soc. (2)*, 61(2):629–640, 2000. (See pgs. 15 and 31.)

[AFR10] N. L. Ackerman, C. E. Freer, and D. M. Roy. On the computability of conditional probability, 2010, math.LO/1005.3014. (See pg. 8.)

[AFR11] N. L. Ackerman, C. E. Freer, and D. M. Roy. Noncomputable conditional distributions. In *Proc. of the 26th Ann. Symp. on Logic in Comp. Sci.* IEEE Press, 2011. (See pg. 8.)

[Ald81] D. J. Aldous. Representations for partially exchangeable arrays of random variables. *J. Multivariate Anal.*, 11(4):581–598, 1981. (See pgs. 99, 108 and 109.)

[Ald85] D. J. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, volume 1117 of *Lecture Notes in Math.*, pages 1–198. Springer, Berlin, 1985. (See pgs. 72 and 98.)

[AM10] A. Asinowski and T. Mansour. Separable d-permutations and guillotine partitions. *Ann. Comb.*, 14(1):17–43, 2010. (See pg. 113.)

[Aus08] T. Austin. On exchangeable random variables and the statistics of large graphs and hypergraphs. *Probab. Surv.*, 5:80–145, 2008. (See pg. 99.)

[BCGL92] S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the theory of average case complexity. *J. Comput. System Sci.*, 44(2):193–219, 1992. (See pg. 39.)

[Ben75] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18:509–517, September 1975. (See pg. 111.)

[Ber06] J. Bertoin. *Random fragmentation and coagulation processes*, volume 102 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2006. (See pgs. 119, 120 and 121.)

[Bos08] V. Bosserhoff. Notions of probabilistic computability on represented spaces. *J.UCS*, 14(6):956–995, 2008. (See pgs. 32, 36, 78 and 89.)

[BP03] V. Brattka and G. Presser. Computability on subsets

of metric spaces. *Theoret. Comput. Sci.*, 305(1-3):43–76, 2003. Topology in computer science (Schloß Dagstuhl, 2000). (See pg. 79.)

[BS94] J. M. Bernardo and A. F. M. Smith. *Bayesian theory*. John Wiley & Sons, 1994. (See pg. 69.)

[BSS07] I. Battenfeld, M. Schröder, and A. Simpson. A convenient category of domains. In *Computation, meaning, and logic: articles dedicated to Gordon Plotkin*, volume 172 of *Electron. Notes Theor. Comput. Sci.*, pages 69–99. Elsevier, Amsterdam, 2007. (See pg. 76.)

[Bun94] W. L. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994. (See pg. 18.)

[CGM+89] D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner, and P. R. Young. Computing as a discipline. *Commun. ACM*, 32(1):9–23, 1989. (See pg. 13.)

[Cha75] G. J. Chaitin. A theory of program size formally identical to information theory. *J. ACM.*, 22:329–340, 1975. (See pg. 30.)

[Daw82] A. P. Dawid. Intersubjective statistical models. In *Exchangeability in probability and statistics (Rome, 1981)*, pages 217–232. North-Holland, Amsterdam, 1982. (See pg. 72.)

[dF31] B. de Finetti. Funzione caratteristica di un fenomeno aleatorio. *Atti della R. Accademia Nazionale dei Lincei, Ser. 6. Memorie, Classe di Scienze Fisiche, Matematiche e Naturali*, 4:251–299, 1931. (See pg. 72.)

[dF37] B. de Finetti. La prévision : ses lois logiques, ses sources subjectives. *Ann. Inst. H. Poincaré*, 7(1):1–68, 1937. (See pg. 72.)

[dF75] B. de Finetti. *Theory of probability. Vol. 2*. John Wiley & Sons Ltd., London, 1975. (See pgs. 73 and 96.)

[DF84] P. Diaconis and D. Freedman. Partial exchangeability and sufficiency. In *Statistics: applications and new directions (Calcutta, 1981)*, pages 205–236. Indian Statist. Inst., Calcutta, 1984. (See pg. 72.)

[DJ08] P. Diaconis and S. Janson. Graph limits and exchangeable random graphs. *Rend. Mat. Appl. (7)*, 28(1):33–61, 2008. (See pg. 99.)

[dMSS56] K. de Leeuw, E. F. Moore, C. E. Shannon, and N. Shapiro. Computability by probabilistic machines. In *Automata studies*, Annals of Math. Studies, no. 34, pages 183–212. Princeton Univ. Press, Princeton, N. J., 1956. (See pg. 31.)

[Eda96] A. Edalat. The Scott topology induces the weak topology. In *11th Ann. IEEE Symp. on Logic in Comput. Sci. (New Brunswick, NJ, 1996)*, pages 372–381. IEEE Comput. Soc. Press, Los Alamitos, CA, 1996. (See pg. 31.)

[Eda97] A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bull. Symbolic Logic*, 3(4):401–452, 1997. (See pg. 25.)

[EH98] A. Edalat and R. Heckmann. A computational model for metric spaces. *Theoret. Comput. Sci.*, 193(1-2):53–73, 1998. (See pg. 28.)

[ES99] M. Escardó and T. Streicher. Induction and recursion on the partial real line with applications to Real PCF. *Theoret. Comput. Sci.*, 210(1):121–157, 1999. (See pg. 92.)

[Esc09] M. Escardó. Semi-decidability of may, must and probabilistic testing in a higher-type setting. *Electron. Notes in Theoret. Comput. Sci.*, 249:219–242, August 2009. (See pgs. 20 and 92.)

[FKP99] N. Friedman, D. Koller, and A. Pfeffer. Structured representation of complex stochastic systems. In *Proc. 15th Nat. Conf. on Artificial Intelligence*, pages 157–164. AAAI Press, 1999. (See pg. 18.)

[FR09a] C. E. Freer and D. M. Roy. Computable de Finetti measures, 2009, math.LO/0912.1072. (See pg. 8.)

[FR09b] C. E. Freer and D. M. Roy. Computable exchangeable sequences have computable de Finetti measures. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *Mathematical Theory and Computational Practice (CiE 2009), Proc. of the 5th Conf. on Computability in Europe*, volume 5635 of *Lecture Notes in Comput. Sci.*, pages 218–231. Springer, 2009. (See pg. 8.)

[FR10] C. E. Freer and D. M. Roy. Posterior distributions are computable from predictive distributions. In *Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2010) (Y. W. Teh and M. Titterington, eds.), JMLR: W&CP 9*, pages 233–240, 2010. (See pgs. 8, 33 and 70.)

[Gác05] P. Gács. Uniform test of algorithmic randomness over a general space. *Theoret. Comput. Sci.*, 341(1-3):91–137, 2005. (See pgs. 28, 34 and 44.)

[GG05] T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Adv. in Neural Inform. Processing Syst. 17*, pages 475–482. MIT Press, Cambridge, MA, 2005. (See pg. 98.)

[GHR10] S. Galatolo, M. Hoyrup, and C. Rojas. Effective symbolic dynamics, random points, statistical behavior, complexity and entropy. *Inform. and Comput.*, 208(1):23–41, 2010. (See pgs. 28, 29, 30, 34, 35, 36 and 46.)

[GMR+08] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *Uncertainty in Artificial Intelligence*, 2008. (See pgs. 8, 15, 20, 21, 22, 92, 93, 94 and 108.)

[Gol67] E. M. Gold. Language identification in the limit. *Information*

and Control, 10(5):447–474, 1967. (See pgs. 40 and 64.)

[Grz57] A. Grzegorczyk. On the definitions of computable real continuous functions. *Fund. Math.*, 44:61–71, 1957. (See pgs. 15 and 28.)

[GSW07] T. Grubba, M. Schröder, and K. Weihrauch. Computable metrization. *Math. Log. Q.*, 53(4-5):381–395, 2007. (See pgs. 28 and 76.)

[GZ89] T. Gonzalez and S.-Q. Zheng. Improved bounds for rectangular and guillotine partitions. *J. Symbolic Comput.*, 7(6):591–610, 1989. (See pg. 113.)

[Har28] R. Hartley. Transmission of information. *Bell System Technical J.*, page 535563, July 1928. (See pg. 50.)

[HLL83] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137, 1983. (See pg. 109.)

[Hof08] P. D. Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Adv. in Neural Inform. Processing Syst. 21*, pages 657–664. MIT Press, Cambridge, MA, 2008. (See pg. 109.)

[Hoo79] D. N. Hoover. Relations on probability spaces and arrays of random variables. Preprint, Institute for Advanced Study, Princeton, NJ, 1979. (See pgs. 99, 108 and 109.)

[HR09] M. Hoyrup and C. Rojas. Computability of probability measures and Martin-Löf randomness over metric spaces. *Inform. and Comput.*, 207(7):830–847, 2009. (See pgs. 28, 31, 32, 34 and 49.)

[HS55] E. Hewitt and L. J. Savage. Symmetric measures on Cartesian products. *Trans. Amer. Math. Soc.*, 80:470–501, 1955. (See pg. 72.)

[HT07] O. Hasan and S. Tahar. Formalization of the standard uniform random variable. *Theor. Comput. Sci.*, 382(1):71–83, 2007. (See pg. 20.)

[Hur02] J. Hurd. A formal approach to probabilistic termination. In V. Carreño, C. Muñoz, and S. Tahar, editors, *TPHOLs*, volume 2410 of *Lecture Notes in Computer Science*, pages 230–245. Springer, 2002. (See pg. 20.)

[Hut07] M. Hutter. On universal prediction and Bayesian confirmation. *Theoret. Comput. Sci.*, 384(1):33–48, 2007. (See pg. 40.)

[Jor10] M. I. Jordan. Bayesian nonparametric learning: Expressive priors for intelligent systems. In R. Dechter, H. Geffner, and J. Halpern, editors, *Heuristics, Probability and Causality: A Tribute to Judea Pearl*. College Publications, 2010. (See pgs. 18 and 107.)

[JP89] C. Jones and G. Plotkin. A probabilistic powerdomain of evaluations. In *Proc. of the Fourth Ann. Symp. on Logic in Comp.*

Sci., pages 186–195. IEEE Press, 1989. (See pg. 20.)

[Kal97] O. Kallenberg. *Foundations of modern probability*. Probability and its Applications (New York). Springer-Verlag, New York, 1997. (See pgs. 117, 118, 120 and 123.)

[Kal02] O. Kallenberg. *Foundations of modern probability*. Springer, New York, 2nd edition, 2002. (See pgs. 41, 73, 74 and 89.)

[Kal05] O. Kallenberg. *Probabilistic symmetries and invariance principles*. Probability and its Applications (New York). Springer, New York, 2005. (See pgs. 71, 72 and 108.)

[Kin78] J. F. C. Kingman. Uses of exchangeability. *Ann. Probability*, 6(2):183–197, 1978. (See pg. 72.)

[Kin93] J. F. C. Kingman. *Poisson processes*, volume 3 of *Oxford Studies in Probability*. The Clarendon Press Oxford University Press, New York, 1993. Oxford Science Publications. (See pg. 110.)

[Kle59] S. C. Kleene. Recursive functionals and quantifiers of finite types. I. *Trans. Amer. Math. Soc.*, 91:1–52, 1959. (See pgs. 15, 25 and 28.)

[Kol33] A. N. Kolmogorov. *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Springer, 1933. (See pgs. 14, 38 and 40.)

[KP97] D. Koller and A. Pfeffer. Object-oriented bayesian networks. In *Proc. 13th Ann. Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 302–313. Morgan Kaufmann, 1997. (See pg. 18.)

[KS09] O. Kiselyov and C. Shan. Embedded probabilistic programming. In W. M. Taha, editor, *Domain-Specific Languages*, volume 5658 of *Lecture Notes in Computer Science*, pages 360–384. Springer, 2009. (See pgs. 20 and 92.)

[KTG+06] C. Kemp, J. Tenenbaum, T. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *Proc. of the 21st Nat. Conf. on Artificial Intelligence*, 2006. (See pgs. 99, 110 and 111.)

[KY76] D. E. Knuth and A. C. Yao. The complexity of nonuniform random number generation. In *Algorithms and complexity (Proc. Sympos., Carnegie-Mellon Univ., Pittsburgh, Pa., 1976)*, pages 357–428. Academic Press, New York, 1976. (See pg. 34.)

[Lau84] S. L. Lauritzen. Extreme point models in statistics. *Scand. J. Statist.*, 11(2):65–91, 1984. (See pg. 72.)

[Lev86] L. A. Levin. Average case complete problems. *SIAM J. Comput.*, 15(1):285–286, 1986. (See pg. 39.)

[Man73] I. Mann. Probabilistic recursive functions. *Trans. Amer. Math. Soc.*, 177:447–467, 1973. (See pg. 34.)

[Man09] V. K. Mansinghka. *Natively Probabilistic Computing*. PhD thesis, Massachusetts Institute of Technology, 2009. (See pg. 98.)

[Maz63] S. Mazur. Computable analysis. *Rozprawy Mat.*, 33:110, 1963. (See pgs. 15 and 28.)

[MM99] C. Morgan and A. McIver. pGCL: formal reasoning for random algorithms. *South African Comput. J.*, 22:14–27, 1999. (See pg. 20.)

[MMS96] C. Morgan, A. McIver, and K. Seidel. Probabilistic predicate transformers. *ACM Trans. Program. Lang. Syst.*, 18(3):325–353, 1996. (See pg. 20.)

[Mül99] N. T. Müller. Computability on random variables. *Theoret. Comput. Sci.*, 219(1-2):287–299, 1999. Computability and complexity in analysis (Castle Dagstuhl, 1997). (See pgs. 80, 81, 82 and 89.)

[MW08] T. Minka and J. Winn. Gates: a graphical notation for mixture models. In *Adv. in Neural Inform. Processing Syst.*, volume 21, 2008. (See pg. 21.)

[Myh71] J. Myhill. A recursive function, defined on a compact interval and having a continuous derivative that is not recursive. *Michigan Math. J.*, 18:97–98, 1971. (See pg. 50.)

[Nie09] A. Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009. (See pg. 29.)

[NS01] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *J. Amer. Stat. Assoc.*, 96:1077–1087(11), 2001. (See pg. 110.)

[O'D11] T. J. O'Donnell. *Productivity and reuse in language*. PhD thesis, Harvard University, 2011. (See pg. 108.)

[Orb10] P. Orbanz. Construction of nonparametric Bayesian models from parametric Bayes equations. In *Adv. in Neural Inform. Processing Syst. 22*, 2010. (See pg. 104.)

[OSW86] D. N. Osherson, M. Stob, and S. Weinstein. *Systems that learn: an introduction to learning theory for cognitive and computer scientists*. MIT Press, Cambridge, MA, USA, 1986. (See pgs. 65 and 66.)

[OSW88] D. N. Osherson, M. Stob, and S. Weinstein. Mechanical learners pay a price for Bayesianism. *J. Symbolic Logic*, 53(4):1245–1251, 1988. (See pgs. 40, 64 and 66.)

[Pea88] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988. (See pgs. 17 and 18.)

[PER89] M. B. Pour-El and J. I. Richards. *Computability in analysis and physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989. (See pgs. 28, 50 and 83.)

[Pfa79] J. Pfanzagl. Conditional distributions as derivatives. *Ann. Probab.*, 7(6):1046–1050, 1979. (See pgs. 38 and 51.)

[Pfe01] A. Pfeffer. IBAL: A probabilistic rational programming language. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence*, pages 733–740. Morgan Kaufmann Publ., 2001. (See

pgs. 20 and 92.)

[Pit96] J. Pitman. Some developments of the Blackwell-MacQueen urn scheme. In *Statistics, probability and game theory*, volume 30 of *IMS Lecture Notes Monogr. Ser.*, pages 245–267. Inst. Math. Statist., Hayward, CA, 1996. (See pg. 103.)

[Plo76] G. D. Plotkin. A powerdomain construction. *SIAM J. Comput.*, 5(3):452–487, 1976. Semantics and correctness of programs. (See pg. 31.)

[Plo77] G. D. Plotkin. LCF considered as a programming language. *Theoret. Comput. Sci.*, 5(3):223–255, 1977. (See pg. 92.)

[PPT08] S. Park, F. Pfenning, and S. Thrun. A probabilistic language based on sampling functions. *ACM Trans. Program. Lang. Syst.*, 31(1):1–46, 2008. (See pgs. 20, 21, 92 and 93.)

[Put85] H. Putnam. *Mathematics, Matter and Method*, volume 1 of *Philosophical Letters*. Cambridge University Press, 1985. (See pgs. 40 and 64.)

[Rad07] A. Radul. Report on the probabilistic language Scheme. Technical Report MIT-CSAIL-TR-2007-059, Massachusetts Institute of Technology, 2007. (See pg. 21.)

[Rao88] M. M. Rao. Paradoxes in conditional probability. *J. Multivariate Anal.*, 27(2):434–446, 1988. (See pg. 38.)

[Rao93] M. M. Rao. *Conditional measures and applications*, volume 177 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1993. (See pg. 37.)

[Rao05] M. M. Rao. *Conditional measures and applications*, volume 271 of *Pure and Applied Mathematics*. Chapman & Hall/CRC, Boca Raton, FL, second edition, 2005. (See pgs. 37 and 38.)

[RD06] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006. (See pg. 18.)

[RKMT07] D. M. Roy, C. Kemp, V. K. Mansinghka, and J. B. Tenenbaum. Learning annotated hierarchies from relational data. In *Adv. in Neural Inform. Processing Syst.*, 2007. (See pgs. 8 and 110.)

[RMGT08] D. M. Roy, V. K. Mansinghka, N. D. Goodman, and J. B. Tenenbaum. A stochastic programming perspective on nonparametric Bayes. Nonparametric Bayesian Workshop, Int. Conf. on Machine Learning, 2008. (See pgs. 15, 93, 99 and 108.)

[RN57] C. Ryll-Nardzewski. On stationary sequences of random variables and the de Finetti's equivalence. *Colloq. Math.*, 4:149–156, 1957. (See pg. 72.)

[Rog87] H. Rogers, Jr. *Theory of recursive functions and effective computability*. MIT Press, Cambridge, MA, second edition, 1987. (See pg. 26.)

[RP02] N. Ramsey and A. Pfeffer. Stochastic lambda calculus and monads of probability distributions. *Proc. of the 29th ACM*

SIGPLAN-SIGACT Symp. on Principles of Program. Lang., pages 154–165, 2002. (See pg. 93.)

[RT09] D. M. Roy and Y. W. Teh. The Mondrian process. In *Adv. in Neural Inform. Processing Syst. 21*, 2009. (See pgs. 8, 99 and 109.)

[Sch95] M. J. Schervish. *Theory of statistics*. Springer Series in Statistics. Springer-Verlag, New York, 1995. (See pg. 48.)

[Sch07] M. Schröder. Admissible representations for probability measures. *Math. Log. Q.*, 53(4-5):431–445, 2007. (See pgs. 31, 32, 43, 78, 79, 81 and 89.)

[Sco75] D. Scott. Data types as lattices. In ⊨*ISILC Logic Conference (Proc. Internat. Summer Inst. and Logic Colloq., Kiel, 1974)*, pages 579–651. Lecture Notes in Math., Vol. 499. Springer, Berlin, 1975. (See pg. 31.)

[SD78] N. Saheb-Djahromi. Probabilistic LCF. In *Mathematical Foundations of Comput. Sci., 1978 (Proc. Seventh Sympos., Zakopane, 1978)*, volume 64 of *Lecture Notes in Comput. Sci.*, pages 442–451. Springer, Berlin, 1978. (See pgs. 20 and 92.)

[Set94] J. Sethuraman. A constructive definition of Dirichlet priors. *Statist. Sinica*, 4(2):639–650, 1994. (See pgs. 98, 101 and 110.)

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proc. I.R.E.*, 37:10–21, 1949. (See pg. 50.)

[Soa87] R. I. Soare. *Recursively enumerable sets and degrees*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. (See pg. 99.)

[Sol64] R. J. Solomonoff. A formal theory of inductive inference II. *Inform. and Control*, 7:224–254, 1964. (See pg. 40.)

[SS06] M. Schröder and A. Simpson. Representing probability measures using probabilistic processes. *J. Complexity*, 22(6):768–782, 2006. (See pgs. 15, 31, 79 and 88.)

[Sto83] L. Stockmeyer. Optimal orientations of cells in slicing floorplan designs. *Inform. and Control*, 57(2-3):91–101, 1983. (See pg. 112.)

[Str05] D. W. Stroock. *An introduction to Markov processes*, volume 230 of *Graduate Texts in Math.* Springer-Verlag, Berlin, 2005. (See pg. 32.)

[Tak08] H. Takahashi. On a definition of random sequences with respect to conditional probability. *Inform. and Comput.*, 206(12):1375–1382, 2008. (See pg. 40.)

[TGG07] Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Proc. of the 11th Conf. on A.I. and Stat.*, 2007. (See pgs. 23, 69, 99 and 110.)

[TJ07] R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. In *Proc. of the 11th Conf. on A.I. and Stat.*, 2007. (See pgs. 69 and 98.)

[Tju74] T. Tjur. *Conditional probability distributions*. Lecture Notes, no.

2. Institute of Mathematical Statistics, University of Copenhagen, Copenhagen, 1974. (See pg. 38.)

[Tju75] T. Tjur. *A Constructive Definition of Conditional Distributions*. Preprint 13. Institute of Mathematical Statistics, University of Copenhagen, Copenhagen, 1975. (See pgs. 38 and 51.)

[Tju80] T. Tjur. *Probability based on Radon measures*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons Ltd., Chichester, 1980. (See pgs. 38, 51 and 52.)

[Tur36] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 42(1):230–265, 1936. (See pgs. 15, 25, 26, 28 and 58.)

[Val84] L. G. Valiant. A theory of the learnable. In *Proc. of the 16th Ann. ACM Symp. on Theory of Comput.*, STOC '84, pages 436–445, New York, NY, USA, 1984. ACM. (See pg. 65.)

[WA87] S. Wasserman and C. Anderson. Stochastic a posteriori blockmodels: Construction and assessment. *Social Networks*, 9(1):1 – 36, 1987. (See pg. 110.)

[Wei89] K. Weihrauch. Constructivity, computability, and computational complexity in analysis. In *Fundamentals of computation theory (Szeged, 1989)*, volume 380 of *Lecture Notes in Comput. Sci.*, pages 480–493. Springer, New York, 1989. (See pg. 28.)

[Wei99] K. Weihrauch. Computability on the probability measures on the Borel sets of the unit interval. *Theoret. Comput. Sci.*, 219(1-2):421–437, 1999. (See pgs. 31, 79, 80, 81 and 89.)

[Wei00a] K. Weihrauch. *Computable analysis: an introduction*. Springer-Verlag, Berlin, 2000. (See pgs. 25 and 28.)

[Wei00b] K. Weihrauch. *Computable analysis: an introduction*. Springer, Berlin, 2000. (See pgs. 75, 76, 77, 78 and 79.)

[WI98] R. L. Wolpert and K. Ickstadt. Simulation of Lévy random fields. In *Practical nonparametric and semiparametric Bayesian statistics*, volume 133 of *Lecture Notes in Statist.*, pages 227–242. Springer, New York, 1998. (See pgs. 23, 69 and 99.)

[WL89] D. F. Wong and C. L. Liu. Floorplan design of VLSI circuits. *Algorithmica*, 4(2):263–291, 1989. (See pg. 112.)

[WZ00] K. Weihrauch and X. Zheng. Computability on continuous, lower semi-continuous and upper semi-continuous real functions. *Theoret. Comput. Sci.*, 234(1-2):109–133, 2000. (See pg. 78.)

[XTYK06] Z. Xu, V. Tresp, K. Yu, and H.-P. Kriegel. Infinite Hidden Relational Models. In *Proc. 22nd Conf. on Uncertainty in Artificial Intelligence*, 2006. (See pg. 110.)

[Yam99] T. Yamakami. Polynomial time samplable distributions. *J. Complexity*, 15(4):557–574, 1999. (See pg. 39.)

[Zhe02] X. Zheng. Recursive approximability of real numbers. *Math. Log. Q.*, 48(suppl. 1):131–156, 2002. (See pgs. 30 and 64.)

[ZL70] A. K. Zvonkin and L. A. Levin. The complexity of finite

objects and the basing of the concepts of information and randomness on the theory of algorithms. *Uspehi Mat. Nauk*, 25(6 (156)):85–127, 1970. (See pg. 40.)