Posterior distributions are computable from predictive distributions

Cameron E. Freer*
Department of Mathematics
Massachusetts Institute of Technology
freer@math.mit.edu

Daniel M. Roy*
Computer Science and A.I. Laboratory
Massachusetts Institute of Technology
droy@csail.mit.edu

Abstract

As we devise more complicated prior distributions, will inference algorithms keep up? We highlight a negative result in computable probability theory by Ackerman, Freer, and Roy (2010) that shows that there exist computable priors with noncomputable posteriors. In addition to providing a brief survey of computable probability theory geared towards the A.I. and statistics community, we give a new result characterizing when conditioning is computable in the setting of exchangeable sequences, and provide a computational perspective on work by Orbanz (2010) on conjugate nonparametric models. In particular, using a computable extension of de Finetti's theorem (Freer and Roy 2009), we describe how to transform a posterior predictive rule for generating an exchangeable sequence into an algorithm for computing the posterior distribution of the directing random measure.

1 Introduction

Bayesian statistics has been revolutionized by the ready availability of vast computing resources that can power a range of numerical techniques, both randomized and deterministic. Probabilistic modeling is an essential tool across all fields of science, and ideas from computer science are playing an ever more central role as models grow in complexity, both in terms of the large size of datasets and the sophistication of the statistical models. Probabilistic programming languages and probabilistic logics are pushing this complexity to new heights; devising generic inference algorithms for the full extent of models expressible in these languages and logics poses a significant challenge. It is critical to develop a theoretical understanding of the possibilities and fundamental limitations of statistical computation. Computable probability theory provides a framework for exploring these questions. We present aspects of this theory in Section 2.

One of the most important computational tasks in Bayesian statistics is the calculation of conditional probabilities, and in particular the calculation of posterior distributions given observed data.

In many settings, computing conditional probabilities is straightforward. For a random variable θ and *discrete* random variable X, the formula

$$\mathbf{P}\{\theta \in A \,|\, X = x\} = \frac{\mathbf{P}\{\theta \in A,\ X = x\}}{\mathbf{P}\{X = x\}} \qquad (1)$$

gives us the conditional probability that $\theta \in A$ given $X = x$, provided that $\mathbf{P}\{X = x\} > 0$. In the case where the conditioned random variable is continuous (and thus $\mathbf{P}\{X = x\} = 0$), the situation is more complicated.

A common situation is where the conditional distribution $\mathbf{P}[X \,|\, \theta]$ has a conditional density $p(x \,|\, \vartheta)$, in which case we say that the likelihood model is *dominated*. Then, the conditional probability is given by Bayes' rule

$$\mathbf{P}[\theta \in A \,|\, X = x] = \frac{\int_A p(x \,|\, \vartheta)\mathbf{P}_\theta(d\vartheta)}{\int p(x \,|\, \vartheta)\mathbf{P}_\theta(d\vartheta)}, \qquad (2)$$

where $\mathbf{P}_\theta$ is the distribution of θ and $\mathbf{P}_\theta(d\vartheta)$ simplifies to $p_\theta(\vartheta)\, d\vartheta$ if θ is absolutely continuous with density p_θ. Like the discrete setting, the dominated continuous setting admits a concrete formula for the conditional probability.

However, in infinite-dimensional nonparametric settings, the likelihood is often not dominated.[1] In these

* The authors contributed equally to this work.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

[1] For a discussion of dominated models and Bayes' theo-

cases, previous work has determined the form of the conditional or has identified underlying finite dimensional structure (e.g., the Chinese restaurant process) that can be used to compute conditional probabilities. One can imagine the difficulty of devising a generic algorithm capable of handling the full extent of known nonparametric models, never mind the infinitude of those yet to be proposed.

Can we build models in such a way that we can always determine the posterior? Orbanz (2010) describes a method for building conjugate nonparametric models with closed form update rules for posterior analysis. Here we study the minimal requirement: when there exists *some* algorithm that computes the posterior distribution to arbitrary precision. Even this is not always possible, as we will see in Section 2.3.

While conditioning is not computable in the general case, we can sometimes exploit additional structure to compute conditional distributions. In particular, we study the setting of *exchangeable* sequences. By de Finetti's theorem, an exchangeable sequence $\{X_i\}_{i\geq 1}$ is conditionally i.i.d. given a latent variable ν called the directing random measure. Using a computable extension of de Finetti's theorem (Freer and Roy 2009) 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 $\{\mathbf{P}[X_{k+1} \,|\, X_{1:k}]\}_{k\geq 1}$.

2 Computable probability theory

Computable probability theory provides a framework for exploring the circumstances in which statistical operations can be performed via *algorithms*. The theory is based on a long tradition of mathematical work in computable analysis studying the computability of continuous functions and higher-order types (see, e.g., (Weihrauch 2000)), and also builds upon work in domain theory and the semantics of programming languages (see, e.g., (Edalat 1997)).

rem, see (Schervish 1995, Thm. 1.31). To see that a Bayes' rule can fail to exist in the nonparametric setting, let $\alpha > 0$, let H be a continuous distribution, and sample a random distribution $F \sim \mathrm{DP}(\alpha H)$ from a Dirichlet process prior (Ferguson 1973). Note that F is an (almost surely) discrete distribution whose atoms are i.i.d. samples from H. Let $X \sim F$ be a sample from F. (Note that an independent sample $G \sim \mathrm{DP}(\alpha H)$ will almost surely have a completely disjoint set of point masses. Hence, the conditional distribution $\mathbf{P}[X \,|\, F]$ is not dominated.) The posterior distribution $\mathbf{P}[F \,|\, X]$ is almost surely concentrated on the measure zero set (under the Dirichlet process prior) of random measures that have a point mass at X. Intuitively, a Bayes' rule reweights the prior to form the posterior, but a measure zero set cannot be reweighted to a positive measure set. Hence, in this case, there is no Bayes' rule.

In examining the computability of statistical operations, we are concerned with tasks that we can perform to arbitrary accuracy, and also with what we cannot do, even approximately. Some results in computable probability theory, such as the computable extension of de Finetti's theorem (described in Section 3) provide explicit algorithms. Other results, such as the noncomputability of conditioning (described in Section 2.3) 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 these results, using techniques from computability theory, to show the impossibility of always computing non-trivial approximations, let alone arbitrarily good ones (see Section 2.3). 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.

2.1 Basic notions

Objects like real numbers and probability measures have, in general, only infinite descriptions. In contrast, on actual computers, we will only be able to interact with these objects in finitary ways. One standard and flexible approach to computable analysis is to represent points in a space using the topology of the space. For example, the standard Euclidean topology on the real line is generated by the countable subbasis $\mathcal{I}_{\mathbb{Q}} := \{(\ell, r) \subseteq \mathbb{R} \,:\, \ell, r \in \mathbb{Q} \text{ and } \ell < r\}$ of open intervals with rational endpoints. Each of these open sets can be thought of as an *approximation* to some real number contained within it. The idea of the computable topological approach is to represent a point by a sequence of better and better approximations. Such a sequence is called a *representation* of the point.

Definition 1 (Computable topological space)**.** Let S be a T_0 topological space with a countable subbasis $\mathcal{S}$, and let $s : \mathbb{N} \to \mathcal{S}$ be an enumeration of $\mathcal{S}$. We say that $(S, \mathcal{S}, s)$ is a *computable topological space* when there is a program that enumerates those triples $a, b, c \in \mathbb{N}$ for which $s(a) \subseteq s(b) \cap s(c)$.

In particular, in a computable topological space, although there may be many names for a given subbasis element, there is a program that can recognize names for identical subbasis elements. Computable topological spaces, as defined here, are instances of the computable T_0 spaces defined in (Grubba, Schröder, and Weihrauch 2007, §3). The following definitions are derived from those in (Weihrauch 2000) and (Schröder 2007).

Example 1. The real numbers $\mathbb{R}$ along with a straightforward enumeration of the open rational intervals $\mathcal{I}_{\mathbb{Q}}$ forms a computable topological space. Likewise, $\mathbb{R}^n$ is a computable topological space under an enumeration of the n-dimensional open boxes with rational corners. The space $\mathbb{R}^\infty$ of real sequences is a computable topological space under an enumeration of the k-dimensional open rational cylinders of the form $(\ell_1, r_1) \times \cdots \times (\ell_k, r_k) \times \mathbb{R}^\infty$, for all $k \geq 1$. Intuitively, a k-dimensional cylinder approximates the first k elements of a sequence to some finite accuracy, and says nothing about the remaining elements of the sequence.

A point in a computable topological space is represented by a sequence of subbasis elements whose intersection is the tightest possible approximation to $\{x\}$.

Definition 2 (Representation of a point)**.** Let $(S, \mathcal{S}, s)$ be a computable topological space, and let $x \in S$ be a point in S. Define $N_x := \{B \in \mathcal{S} : x \in B\}$ to be the set of subbasis elements containing x. A *representation* of x is a sequence $a_1, a_2, \ldots$ of s-encodings of subbasis elements containing x for which $\bigcap_{i=1}^\infty s(a_i) = \bigcap N_x$.

When S is a T_1 topological space (e.g., those in Example 1) the above representation of x reduces to $\{x\} = \bigcap_{i=1}^\infty s(a_i)$.

Definition 3 (Computable point)**.** We say that a point $x \in S$ is *computable* when there is a program that, on input i, outputs a_i, for some representation $(a_i)_{i=1}^\infty$ of x.

We may also think of a computable point as being given by a program that (on empty input) outputs an infinite stream whose ith term is a_i.

Example 2. A real number $\alpha \in \mathbb{R}$ is computable when there is a computable sequence of rational intervals that converges to $\{\alpha\}$. Equivalently, one can take the sequence to be rapidly converging; e.g., one can show that α is computable if and only if there is a program that, on input $k \geq 1$ outputs a rational $q_k \in \mathbb{Q}$ satisfying $|\alpha - q_k| < 2^{-k}$. A real sequence $\{\alpha_i\}_{i \geq 1} \in \mathbb{R}^\infty$ is represented by a sequence of cylinders that eventually approximates every element to arbitrary accuracy. Equivalently, one might require that the kth term of the representation is a k-cylinder that specifies the first k elements $\alpha_1, \alpha_2, \ldots, \alpha_k$ to precision 2^{-k} (and says nothing about the remaining terms).

A computable function maps representations of input points to representations of output points.[2] The essential property is that when a program has produced a finite portion of its output (e.g., one term of a representation sequence), it has done so in finite time, having consumed a finite, but unbounded, portion of its input.

Definition 4 (Computable function)**.** Let $(S, \mathcal{S}, s)$ and $(T, \mathcal{T}, t)$ be computable topological spaces. Let $f : S \to T$ be a continuous function. We say that the function f is *computable*[3] when there is a program that, given as input a representation of a point $x \in S$, outputs a representation of the point $f(x)$.

Let $f : S \to T$ and $g : T \to U$ be computable functions. Then their composition $g \circ f : S \to U$ is also a computable function. Also note that a computable function maps computable points to computable points (as can seen by wrapping the transformation describing the function around the program generating the computable point).

In order to study the computability of operations like conditioning, we must choose suitable notions of computability for distributions and random variables. Following the same pattern, we will make the space $\mathscr{P}(S)$ of (Borel) probability measures on S into a computable topological space, and then perform computations on measures via their representations as points. Before we specify an appropriate topology for $\mathscr{P}(S)$, we first define the notion of a computable random variable, as it will suggest an appropriate topology to use for defining computable measures.

2.2 Computable random variables

Intuitively, random variables map an input source of randomness to an output, inducing a distribution on the output space. From the perspective of computability, there are various equivalent sources of randomness. Here we will use a sequence of independent fair coin flips, which generates the same rich class of computable distributions as that generated by more sophisticated sources, such as uniform random variables.

The space $\{0,1\}^\infty$ of infinite binary sequences is a computable topological space (under a straightforward enumeration of the product topology). We will use this space as a source of randomness, and so we will consider it as a probability space where the distribution is that of an i.i.d. sequence of fair coins.

Definition 5 (Computable random variable)**.** Let $(S, \mathcal{S}, s)$ be a computable topological space. Then an

[2] The input representation is typically infinite. One way to formalize this is via oracle Turing machines. Another approach is to work directly with machines that handle infinite streams, as in the Type-two Theory of Effectivity (TTE) described in (Weihrauch 2000).

[3] Computable functions $S \to T$ can equivalently be viewed as the computable points in the computable topological space of continuous functions $S \to T$ under the compact-open topology (Weihrauch 2000, Lem. 6.1.7).

S-valued random variable $\xi : \{0,1\}^\infty \to S$ is computable[4] when there is a program that, given as input a representation of a bit tape $\omega \in \{0,1\}^\infty$, outputs a representation of the point $\xi(\omega)$ for all but a measure zero subset of bit tapes ω.

As with computable functions, the essential property 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 at some point, it consumes its entire 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 3. Given a bit tape $\omega \in \{0,1\}^\infty$, for each $k \geq 1$ set $x_k(\omega) := \sum_{i=1}^{k} \omega_i 2^{-i}$. Define X_k to be the rational interval $(x_k(\omega),\ x_k(\omega) + 2^{-k})$. Note that, for every ω, we have $|x_k(\omega) - x_{k+1}(\omega)| \leq 2^{-(k+1)}$, and so $\lim_k x_k(\omega)$ exists. Thus the sequence of rational intervals $(X_k(\omega))_{k=1}^\infty$ is a representation of the real number $\lim_k x_k(\omega)$, and the distribution of the real number it defines (as $\omega \in \{0,1\}^\infty$ varies according to fair coin flip measure) is uniform on $[0,1]$. Because each interval $X_k(\omega)$ is computed using only finitely many bits of ω, the function defined by $\omega \mapsto (X_k(\omega))_{k=1}^\infty$ constitutes a computable random variable.

It is also possible for a computable random variable to describe an infinite sequence of values, even though infinitely many bits of randomness are already needed for the first element of the sequence. This is accomplished by dovetailing.

Example 4. We extend the example of a uniform random variable to an i.i.d.-uniform sequence. First divide up[5] ω into a countable sequence of disjoint subsequences $(\pi_n(\omega))_{n=1}^\infty$. For $k \geq 1$, let the random rational intervals X_k be as in Example 3, and for $n < k$ define the random rational interval $y_{n,k}(\omega) := X_k(\pi_n(\omega))$. Finally, for each $k \geq 1$ define the random k-dimensional cylinder $Y_k(\omega) := y_{1,k}(\omega) \times y_{2,k}(\omega) \times \cdots \times y_{k,k}(\omega) \times \mathbb{R}^\infty$. As before, for each $n \geq 1$, the sequence $\big(y_{n,k}(\omega)\big)_{k=1}^\infty$ is a representation of the real $\lim_k x_k(\pi_n(\omega))$, and the sequence of cylinders $\big(Y_k(\omega)\big)_{k=1}^\infty$ is a representation of the sequence of these reals. Because the subsequences $(\pi_n(\omega))_{n=1}^\infty$ are disjoint, the random reals $\lim_k x_k(\pi_n(\omega))$ are independent, and by Example 3, uniformly distributed. Because each cylinder $Y_k(\omega)$ is computed using only finitely many bits of ω, the function defined by $\omega \mapsto \big(Y_k(\omega)\big)_{k=1}^\infty$ is a computable sequence-valued random variable.

In Examples 3 and 4, the computable random variable has been defined by a program that outputs a representation on *every* bit tape $\omega \in \{0,1\}^\infty$. We now describe a procedure that sometimes fails to output a representation, but only for a measure zero subset of $\{0,1\}^\infty$.

Example 5. Let $\alpha \in [0,1]$ be a computable real, as in Example 2. Sample $x(\omega)$ from a uniform random variable using bit tape $\omega \in \{0,1\}^\infty$. With probability one, $x \neq \alpha$, and using the computable sequence defining α we can eventually output 1 when $x < \alpha$ and 0 when $x > \alpha$. This procedure constitutes a computable random variable, and so the Bernoulli(α) distribution is computable. Note that this procedure fails to output a representation[6] when $x(\omega) = \alpha$, which occurs when ω represents the binary expansion of α.

What can we learn from observing the behavior of a computable random variable? Consider a computable random variable ξ on a computable topological space $(S, \mathcal{S}, s)$ and let $\mathbf{P}_\xi$ be its distribution. Note that if a program computing ξ outputs an integer n encoding a particular subbasis element $N = s(n) \in \mathcal{S}$, having read only the first k bits $\omega_1 \cdots \omega_k$ of its bit tape, then $\mathbf{P}_\xi(A) \geq 2^{-k}$ for every subbasis element $A \in \mathcal{S}$ for which $A \supseteq N$, because for every bit tape beginning with $\omega_1 \cdots \omega_k$, the program also outputs n. Given a representation of ξ, we can record, for each rational interval, those finite bit tape prefixes that are mapped to subsets, thereby tabulating arbitrarily good rational lower bounds on the quantities $\mathbf{P}_\xi(A)$ for all $A \in \mathcal{S}$.

In fact, lower bounds such as these, on a somewhat richer class of open sets, are precisely what is needed to sample from a distribution, and define a natural subbasis for the *weak topology* on the space of probability measures.

Lemma 1 (Schröder (2007, Lem. 3.2))**.** *Let $(S, \mathcal{S}, s)$ be a computable topological space, and let $\mathcal{A}_\mathcal{S}$ be the closure of $\mathcal{S}$ under finite unions and finite intersections. Then the collection of sets of the form*

$$\{\mu \in \mathscr{P}(S) \,:\, \mu(A) > q\}, \tag{3}$$

[4] Computable S-valued random variables can likewise be viewed as the computable points in a computable topological space, which can be constructed as a subspace of the function space of continuous maps from $\{0,1\}^\infty$ to a suitable augmentation of S (Schröder 2007, §3.3).

[5] Set $\pi_n(\omega) := \omega_{\langle n,1\rangle}\omega_{\langle n,2\rangle}\omega_{\langle n,3\rangle}\cdots$, where $\langle n,k\rangle := \frac{(n+k-1)(n+k-2)}{2} + k$ is a pairing function that bijectively maps pairs of positive integers to positive integers.

[6] It is essential that computable random variables be allowed to fail on a measure zero set, or else natural examples like this one would have to be ruled out. For example, one can show that for any computable Bernoulli(2/3) random variable, there is a bit tape on which it does not halt.

where $A \in \mathcal{A}_S$ and $q \in \mathbb{Q}$, forms a subbasis for the weak topology on the space $\mathscr{P}(S)$ of probability measures on S.

For a computable topological space S, we will use the subbasis given by (3) to turn $\mathscr{P}(S)$ into a computable topological space. For our purposes, Lemmas 2 and 3 justify this choice of topology.[7]

Lemma 2 (Schröder (2007, Prop. 4.3))**.** *Let ξ be a random variable on a computable topological space $(S, \mathcal{S}, s)$. There is a program that takes as input a representation of ξ and outputs a representation of the distribution of ξ. In particular, the distribution of a computable random variable is computable.*

Lemma 3 (Schröder (2007, Prop. 4.3))**.** *Let μ be a distribution on a computable topological space $(S, \mathcal{S}, s)$. There is a program that takes as input a representation of μ and outputs a representation of a random variable with distribution μ.*

Note that from a representation of μ, we can also compute a representation of an i.i.d.-μ sequence, as in Example 4.

2.3 Conditional distributions

Conditioning is a fundamental operation in statistics; it is the process by which a probabilistic model is updated to include new observations. The central challenge facing probabilistic programming language designers is to build inference algorithms that cover as wide a range of scenarios as possible.

Let ξ be a random variable in a computable topological space $(S, \mathcal{S}, s)$, and let η be a random variable in $\mathbb{R}^k$ with distribution $\mathbf{P}_\eta$. A measurable function $\phi : \mathbb{R}^k \to \mathscr{P}(S)$ is called a *version*[8] *of the conditional distribution* $\mathbf{P}[\xi \,|\, \eta]$ when it satisfies

$$\mathbf{P}\{\xi \in A, \eta \in B\} = \int_B \phi(t)(A)\, \mathbf{P}_\eta(dt), \qquad (4)$$

for all measurable sets $A \subseteq S$ and $B \subseteq \mathbb{R}^k$.

Definition 6 (Computable conditional distributions)**.** We say that a version ϕ of the conditional distribution $\mathbf{P}[\xi \,|\, \eta]$ is computable[9] when there is a program that, given as input a representation of a point $t \in \mathbb{R}^k$, outputs a representation of the measure $\phi(t)$, for $\mathbf{P}_\eta$-almost all inputs t.

Note that any computable version ϕ is $\mathbf{P}_\eta$-almost everywhere continuous.

Given an observation of a computable integer-valued random variable, we can compute conditional distributions using Eq. (1). However, the class of computable distributions is not closed under conditioning on computable *continuous* random variables.

Theorem 1 (Noncomputability of conditioning (Ackerman, Freer, and Roy 2010))**.** *There is a pair of computable random variables ξ, η in $[0, 1]$ for which there is an $\mathbf{P}_\eta$-almost everywhere continuous version of the conditional distribution $\mathbf{P}[\xi \,|\, \eta]$, but no version of $\mathbf{P}[\xi \,|\, \eta]$ is computable.*

The proof reduces the halting problem to an expression involving conditional probabilities. If there were a generic algorithm for conditioning, there would then be an algorithm for solving the halting problem, a contradiction.

Theorem 1 implies that is it impossible to compute exact conditional distributions. In fact, the result can be strengthened to show that there is no algorithm that, on every input, outputs some nontrivial finite approximation to the conditional distribution.[10]

Hence a challenge for computable probability theory is to characterize broadly applicable circumstances where conditioning (and therefore Bayesian analysis) *is* computable.[11]

3 Computable de Finetti measures

A *random probability measure on* $\mathbb{R}$ is a random variable $\nu : \{0,1\}^\infty \to \mathscr{P}(\mathbb{R})$. The distribution of ν is a point in the space $\mathscr{P}(\mathscr{P}(\mathbb{R}))$.

Example 6. Let $u : \{0,1\}^\infty \to [0,1]$ be a uniform random variable. Then $\nu := u\delta_1 + (1-u)\delta_0$ is a random Bernoulli measure whose parameter is uniformly distributed. The distribution of ν is a distribution

[7] Furthermore, it can be shown that the representation of measures as points in this space is complete among representations for which the integral operator (for lower semicontinuous and bounded countinuous functions) is computable (Schröder 2007, Prop. 3.6). Hence expectation of bounded random variables and marginalization are computable operations.

[8] Any two measurable functions ϕ_1, ϕ_2 satisfying (4) need only agree $\mathbf{P}_\eta$-almost everywhere.

[9] Computable versions of conditional distributions are computable points in a computable topological space, constructed as in the random variable case.

[10] Furthermore, for every algorithm that only sometimes outputs approximations, and every input on which it does output some valid finite approximation, there is another input representing the same pair of random variables, on which it outputs nothing.

[11] Note that it will not suffice to restrict to a small class of primitives, as long as recursion remains: the random variables ξ and η are defined in terms of simple random variables (uniform, Bernoulli, and geometric) using recursion. Removing recursion, however, would destroy the source of much of the power and flexibility of probabilistic programming languages.

on probability measures that is concentrated on the Bernoulli measures.

Let $X = \{X_i\}_{i\geq 1}$ be an infinite 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 2 (de Finetti's theorem (Hewitt and Savage 1955))**.** *Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence of real random variables. There is an (a.s. unique) random probability measure ν on $\mathbb{R}$ such that X is conditionally i.i.d. with respect to ν:*

$$\mathbf{P}[X_1 \in B_1,\ X_2 \in B_2,\ \ldots \mid \nu] = \textstyle\prod_{i\geq 1} \nu(B_i) \quad a.s.,$$

for Borel sets $B_i \subseteq \mathbb{R}$.

The random measure ν is called the *directing random measure*. Its distribution μ (a measure on probability measures) is called the *mixing measure* or the *de Finetti measure*. Note that ν is, in general, an infinite dimensional object. However, in many settings, the random measure corresponds to a particular member of a parametrized family of distributions, and in this case, the mixing measure corresponds to a distribution on parameters.[12]

Example 7. Consider the sequence $\{Y_k\}_{k\geq 1}$ where, for each $k \geq 1$, we sample $Y_k \sim \mathcal{N}(\frac{1}{k}\sum_{i=1}^{k-1} Y_i,\ 1 + \frac{1}{k})$. 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 $\mathcal{N}(0, 1)$ distribution. The mixing measure can be derived from $\mathbf{P}_Z$, as follows: Let $M : \mathbb{R} \to \mathscr{P}(\mathbb{R})$ be the map that takes a real m to the Gaussian $\mathcal{N}(m, 1)$. Then the mixing measure μ is given by $\mu(B) = \mathbf{P}_Z(M^{-1}(B))$, where B is a (Borel measurable) subset of $\mathscr{P}(\mathbb{R})$ 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.

The classical de Finetti's theorem shows that the distribution of an exchangeable sequence is completely characterized by its mixing measure. A natural question is whether a computable exchangeable sequence necessarily has a computable mixing measure, and furthermore whether it is possible to recover a representation of the mixing measure from a representation of the sequence distribution. The following computable extension[13] of de Finetti's theorem shows that this is in fact possible.

Theorem 3 (Computable de Finetti (Freer and Roy 2009))**.** *Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence of random variables with distribution χ, and let ν be its directing random measure (with distribution μ). Then there is a program that computes a representation of the mixing measure μ from a representation of the sequence distribution χ, and vice versa.*

Example 8. Consider the sequence $\{X_k\}_{k\geq 1}$ of random variables induced by the following urn scheme (Blackwell and MacQueen 1973) whose combinatorial structure is known as the Chinese restaurant process (Aldous 1985). Let $\alpha > 0$ be a computable real and let H be a computable distribution on $\mathbb{R}$. For $k \geq 0$, sample $X_{k+1} \sim \frac{1}{k+\alpha}\sum_{i=1}^{k} \delta_{X_i} + \frac{\alpha}{k+\alpha} H$. The sequence $\{X_k\}_{k\geq 1}$ is exchangeable and its directing random measure is known to be a Dirichlet process (Blackwell and MacQueen 1973). By Lemma 2, the distribution of the exchangeable sequence is computable from the sampler. The computable de Finetti theorem can automatically transform the sequence distribution into its mixing measure, which in this case is the "Dirichlet process prior"[14] with parameter αH. Coming full circle, by Lemma 3, we can use this representation to sample a Dirichlet process F, and in turn repeatedly sample observations $\hat{X}_k \sim F$ from the Dirichlet process, generating an exchangeable sequence $\{\hat{X}_k\}_{k\geq 1}$ of reals equal in distribution to the

[12] The mixing measure is often interpreted as a prior in the Bayesian setting. However, it is important to reiterate that it is uniquely pinned down by the distribution of the exchangeable sequence, which itself may be described without reference to any such prior.

[13] The directing random measure is classically given by an explicit limiting expression. However, without a computable handle on the rate of convergence of the limit, it cannot be used directly to compute the de Finetti measure. Nevertheless, it is possible to reconstruct the de Finetti measure using the moments of a set of derived random variables. For more details, see (Freer and Roy 2009).

[14] We note the following fact about the computability of the stick breaking representation (Sethuraman 1994), which is the list of atoms (and their masses) that comprise the Dirichlet process. When two computable reals are not the same, we can eventually recognize this, but when they are the same, we cannot recognize this. Likewise, given a representation of a distribution that is known to be discrete, although we can list the atoms (as points in $\mathbb{R}^\infty$), it is not possible to recover their masses. Therefore it is not possible to computably transform a Dirichlet process to its stick-breaking representation. Thus, even in settings where the computable de Finetti theorem tells us that the directing random measure is computably distributed, this may (as in Example 7) or may not (as described here) be true of other random variables that render the sequence conditionally independent.

original sequence $\{X_k\}_{k\geq 1}$ induced by the Blackwell-MacQueen urn scheme.

As a practical matter, the particular transformation from an exchangeable sequence to its de Finetti measure given by the proof of Theorem 3 is sometimes rather inefficient. It is an open challenge to identify circumstances where the de Finetti measure can be computed efficiently.

4 Posterior analysis of exchangeable sequences

Let $X = \{X_i\}_{i\geq 1}$ be an exchangeable sequence. Even if the distribution of X is computable, $\mathbf{P}[X_{k+1}|X_{1:k}]$ is not necessarily computable. However, in most cases, our knowledge of an exchangeable sequence is, in fact, precisely of this form: a 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 prediction rule to subsequently sample from the conditional distribution of X_{k+2} given $X_{1:k+1}$, and so on, hallucinating an entire infinite exchangeable sequence given the original prefix. The following result shows that the ability to hallucinate 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 4. *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 $x_{1:j} \in \mathbb{R}^j$. By Lemma 2, 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 and Lemma 3, 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) $\{\mathbf{P}[X_{k+1} \,|\, X_{1:k}]\}_{k\geq 0}$. By Lemma 3, given observed values $x_{1:j}$ for a prefix $X_{1:j}$, we can sample $\hat{X}_{j+1} \sim \mathbf{P}[X_{k+1} \,|\, X_{1:k} = x_{1:k}]$. 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 $X_{1:j} = x_{1:j}$. By Lemma 2, we can compute the conditional distribution of the exchangeable sequence.

Finally, by Theorem 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, and not some other parametrization 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 4 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 9. Recall Example 8, defining a Dirichlet process. Note that the Blackwell-MacQueen prediction rule satisfies the hypotheses of Theorem 4. The proof of Theorem 4 (if implemented as code) automatically transforms the prediction rule into the (computable) posterior distribution

$$\{x_i\}_{i\leq k} \mapsto \mathrm{DP}(\alpha H + \textstyle\sum_{i=1}^{k} \delta_{x_i}). \tag{5}$$

Posterior computation for many other species sampling models (Pitman 1996) 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.

In fact, 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.[15]

[15] Exchangeable structure need not be evident for Theorem 4 to apply. Let G be a distribution on $\mathscr{P}(\mathbb{R})$; 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 4 implies that if we introduce nuisance variables $X_2, X_3, \dots$ that are themselves independent draws from F, then $\mathbf{P}[F \,|\, X]$ is computable if the sequence $\mathbf{P}[X_2 \,|\, X], \mathbf{P}[X_3 \,|\, X, X_2], \dots$ is computable. So even though the model only invokes a single sample from F, the abil-

5 Related work

Orbanz (2010) proves a version of Kolmogorov's extension theorem for families of conditional distributions, providing a new 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 $\mathbf{P}[X_{k+1} \mid \sum_{i=1}^{k} T(X_i)]$, 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 1. *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 $\mathbf{P}[\nu \mid \sum_{i=1}^{k} T(X_i)]$ for $k \geq 1$ is computable if and only if the sequence of conditional distributions $\mathbf{P}[X_{k+1} \mid \sum_{i=1}^{k} T(X_i)]$, for $k \geq 1$, and the sufficient statistic T are computable.*

Corollary 1 and Theorem 4 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.

However, 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. But computable distributions do 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.

ity to do posterior analysis on F given X is linked to our ability to sample the sequence $X_2, X_3, \ldots$ given X.

Acknowledgements

The authors would like to thank Peter Orbanz, Joshua Tenenbaum, and the anonymous reviewers for helpful suggestions, and Leslie Kaelbling and Sinead Williamson for comments on a draft.

References

N. L. Ackerman, C. E. Freer, and D. M. Roy. On the computability of conditional probability. Preprint, 2010.

D. J. Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983*, Lecture Notes in Math., vol. 1117, pages 1–198. Springer, Berlin, 1985.

D. Blackwell and J. B. MacQueen. Ferguson distributions via Pólya urn schemes. *Ann. Statist.*, 1:353–355, 1973.

A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bull. Symbolic Logic*, 3(4):401–452, 1997.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Ann. Statist.*, 1:209–230, 1973.

C. E. Freer and D. M. Roy. Computable exchangeable sequences have computable de Finetti measures. In *Proc. of the 5th Conf. on Computability in Europe*, volume 5635 of *Lecture Notes in Comput. Sci.*, pages 218–231. Springer, 2009.

T. Grubba, M. Schröder, and K. Weihrauch. Computable metrization. *Math. Logic Q.*, 53(4-5):381–395, 2007.

E. Hewitt and L. J. Savage. Symmetric measures on Cartesian products. *Trans. Amer. Math. Soc.*, 80:470–501, 1955.

P. Orbanz. Construction of nonparametric Bayesian models from parametric Bayes equations. In *Adv. in Neural Inform. Processing Syst. 22*, 2010.

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.

M. J. Schervish. *Theory of statistics*. Springer Series in Statistics. Springer-Verlag, New York, 1995.

M. Schröder. Admissible representations for probability measures. *Math. Logic Q.*, 53(4-5):431–445, 2007.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statist. Sinica*, 4(2):639–650, 1994.

K. Weihrauch. *Computable analysis*. Springer-Verlag, Berlin, 2000.