ON COMPUTABLE REPRESENTATIONS OF EXCHANGEABLE DATA

NATHANAEL L. ACKERMAN[1], JEREMY AVIGAD[2], CAMERON E. FREER[3],
DANIEL M. ROY[4], AND JASON M. RUTE[5]

ABSTRACT. An array of data is *exchangeable* when its distribution does not depend on the order of the rows/columns. In hand-engineered systems, exchangeability is often exploited to accelerate probabilistic inference, yet few probabilistic programming languages beyond the Church family have any special support for exchangeability. Mathematically, no special support would seem to be necessary: given exchangeable arrays of data of unbounded size, results by de Finetti, Aldous, and Hoover lead to stateless representations of the arrays that expose the key conditional independences and symmetries. We investigate the computability of the representations, and whether they can themselves be computed given a representation of an exchangeable sequence or array. We give both positive and negative results, and discuss the consequences for the design of probabilistic programming languages.

1. Introduction

Exchangeable sequences are models for homogeneous data sets and serve as building blocks from which more interesting dependency structures in statistical models are produced. A sequence of random variables is *exchangeable* when its distribution does not depend on the ordering of its elements. In programming terms, exchangeable sequences behave as if they were the result of a repeated evaluation of a closure in a probabilistic language, where the closure itself contains a random variable. Exchangeability therefore licenses a programmer or compiler to commute and even remove expressions. Few languages have any special support for exchangeable expressions, with the notable exception of Church [GMRBT08] and its descendants. Church possessed exchangeable random primitives (XRPs), abstract data types that exposed the operations necessary for a compiler to construct advanced Markov chain Monte Carlo algorithms that take advantage of the additional laws that exchangeable sequences satisfy. For further details on XRPs, see [AFR16a] and the references therein. A fundamental question for probabilistic programming is whether or not support for exchangeability is in some sense necessary on the grounds of efficiency or even computability. Below we give partial answers to this question for sequences, but also look ahead to more complicated structures beyond sequences, namely arrays, where the story is much more complicated. In particular, we describe several results on the computable content of the Aldous–Hoover theorem, and its consequences for probabilistic programming languages.

2. Exchangeable sequences and the Computable de Finetti Theorem

A sequence of random variables is **exchangeable** when its joint distribution is invariant to arbitrary permutations of the variables.

Theorem 1 (de Finetti). *Suppose that $\langle X_i \rangle_{i\in\mathbb{N}}$ is an exchangeable sequence of $\{0,1\}$-valued random variables. Then there is a measurable function $f\colon [0,1]^2 \to \{0,1\}$ such that $\langle X_i \rangle_{i\in\mathbb{N}}$ and $\langle f(\zeta_\emptyset, \zeta_i)\rangle_{i\in\mathbb{N}}$ have the same distribution, where $(\zeta_\emptyset, \langle \zeta_i \rangle_{i\in\mathbb{N}})$ is a collection of independent uniform $[0,1]$-valued random variables.*

From a programming languages perspective, one may view de Finetti's theorem as saying that every thunk that maintains state but whose return values are exchangeable is actually equivalent in terms of its observable behavior to a thunk that does not maintain state. We may think about the former representation as sequential (as state requires the separate evaluates to communicate) and the latter as parallel. For more details on de Finetti's theorem, see [Kal05, Ch. 1].

[1] Harvard University; [2] Carnegie Mellon University; [3] Gamalon Machine Intelligence and Borelian Corporation; [4] University of Toronto; [5] Penn State University.

E-mail addresses: nate@math.harvard.edu, avigad@cmu.edu, freer@borelian.com, droy@utstat.toronto.edu, jmr71@math.psu.edu.

Given a computable exchangeable sequence, it is natural to ask when a computable representation exists. Furthermore, one may ask whether a computable representation can be uniformly found in terms of the sequence, thereby uniformly changing a sequential algorithm into a parallel one. In fact, this is possible, as was shown in [FR12]; see also [FR10]. Recall that a measurable function is *a.e. computable* if it is computable on a subset of its domain of full measure.

Theorem 2 (Freer and Roy [FR12])**.** *Given an computable exchangeable sequence $\langle X_i \rangle_{i\in\mathbb{N}}$ there is an a.e. computable function $f\colon [0,1]^2 \to \{0,1\}$ such that for any collection of independent uniform $[0,1]$-valued random variables $(\zeta_\emptyset, \langle \zeta_i \rangle_{i\in\mathbb{N}})$, $\langle X_i \rangle_{i\in\mathbb{N}}$ and $\langle f(\zeta_\emptyset, \zeta_i) \rangle_{i\in\mathbb{N}}$ have the same distribution.*

XRPs allow programmers to use mutation within a procedure as long as the sequence of return values is exchangeable. The computable de Finetti theorem shows a sense in which XRPs are not strictly necessary, and the functionality that they provide could be obtained automatically within a language without mutation by a (slow, but uniformly computable) program transformation. However, exchangeable representations of certain stochastic processes are believed to be more efficient than their conditionally i.i.d. counterparts. The complexity of computable de Finetti remains an open problem. Were XRPs necessary on efficiency grounds, this would imply that probabilistic programming languages would in a sense need to be aware of probabilistic symmetries.

3. Exchangeable arrays, the Aldous–Hoover Theorem, and Noncomputability

For the remainder of this abstract we consider arrays, where the situation is more complicated than for sequences. A two-dimensional array of random variables is **jointly exchangeable** (or simply, *exchangeable*) when its joint distribution is unchanged by simultaneously permuting the rows and columns by the same permutation of $\mathbb{N}$.

The Aldous–Hoover theorem provides a representation very similar to that in de Finetti's theorem. We begin with a definition.

Definition 3. Given a measurable function $f\colon [0,1]^4 \to \{0,1\}$, the **Aldous–Hoover distribution induced by** f is defined to be the distribution of $\langle f(\zeta_\emptyset, \zeta_i, \zeta_j, \zeta_{\{i,j\}}) \rangle_{i,j\in\mathbb{N}}$, where $(\zeta_\emptyset, \langle \zeta_i \rangle_{i\in\mathbb{N}}, \langle \zeta_{\{i,j\}} \rangle_{i<j\in\mathbb{N}})$ is some collection of independent uniform $[0,1]$-valued random variables.

Theorem 4 (Aldous, Hoover)**.** *Suppose $\langle X_{i,j} \rangle_{i,j\in\mathbb{N}}$ is a jointly exchangeable array of $\{0,1\}$-valued random variables. Then the distribution of $\langle X_{i,j} \rangle_{i,j\in\mathbb{N}}$ is that induced by some measurable function $f\colon [0,1]^4 \to \{0,1\}$.*

We say that f is an **Aldous–Hoover representation** of the random array $\langle X_{i,j} \rangle_{i,j\in\mathbb{N}}$ or of its distribution. For more details on the Aldous–Hoover theorem, see [Kal05, Ch. 7]. We may now ask similarly: Given an exchangeable array $\langle X_{i,j} \rangle_{i,j\in\mathbb{N}}$ whose distribution is computable, does it have a computable Aldous–Hoover representation? Here the answer, presented in [AFR16b], is negative.

Theorem 5 (Ackerman, Freer, and Roy [AFR16b])**.** *There is a exchangeable array $\mathbf{X} = \langle X_{i,j} \rangle_{i<j}$ with the following properties:*

- *The distribution of $\mathbf{X}$ is computable.*
- *There is an almost everywhere continuous function $f\colon [0,1]^4 \to \{0,1\}$ which is an Aldous–Hoover representation of $\mathbf{X}$.*
- *If $g\colon [0,1]^4 \to \{0,1\}$ is any a.e. continuous Aldous–Hoover representation of $\mathbf{X}$ then g computes the halting problem $\mathbf{0}'$.*

Below, in Theorem 10 due to Avigad, Freer, Roy, and Rute [AFRR16], we will see a positive answer for a somewhat weaker notion of computability on a natural subclass of exchangeable arrays. This in turn suggests some ways in which probabilistic programming languages might benefit from alternative representational choices.

The fact that the Aldous–Hoover theorem is not uniformly computable suggests that a 2-dimensional array analogue to XRPs provides a user with a provably richer interface for expressing probabilistic symmetries than the language without this array XRP. Given the important role of exchangeability in efficient inference, it seems likely that languages will need to provide some mechanism for users to express the invariances their models possess.

4. Mixing measures

A standard result of ergodic theory implies that every exchangeable array is a mixture of ergodic ones. The computable de Finetti theorem may be viewed as saying that given an computable exchangeable sequence, the "mixing measure" on the space of measures is computable, i.e., one can computably decompose the distribution of the exchangeable sequence into a mixture of product measures. It is then natural to ask if the same result holds for exchangeable arrays; in fact it does hold, as a consequence of (a slight extension of) Theorem 2.

Corollary 6 (Ackerman, Freer, and Roy [AFR16b])**.** *Let μ be a computable measure on the space $\{0,1\}^{\mathbb{N}^2}$ that is the distribution of some exchangeable array. Then there is a computable measure ν on the space of ergodic measures on $\{0,1\}^{\mathbb{N}^2}$ such that μ is the ν-mixture of the ergodic measures.*

Unfortunately, knowing that a computable exchangeable array has a computable mixing measure does not get us much closer to a computable Aldous–Hoover representation. One might hope for such a representation when the mixing measure is a distribution over ergodic measures, each of which has a computable Aldous–Hoover representation itself. However, even this is not enough in general.

Theorem 7 (Ackerman, Freer, and Roy [AFR16b])**.** *There is an exchangeable array $\mathbf{X} = \langle X_{i,j}\rangle_{i<j}$ with the following properties:*

- *The distribution of $\mathbf{X}$ is computable.*
- *The distribution of $\mathbf{X}$ is a discrete mixture of measures $\langle \mu_i\rangle_{i\in\mathbb{N}}$ such that*
  - *each μ_i is assigned a rational weight, and*
  - *each μ_i has an Aldous–Hoover representation which is a step function with at most 6 steps, each of whose steps has rational endpoints (in particular, such a representation is computable).*
- *If $\langle U_i\rangle_{i\in\mathbb{N}}$ is a list of a.e. continuous Aldous–Hoover representations of the measures $\langle \mu_j\rangle_{j\in\mathbb{N}}$ (in an arbitrary order) then the collection $\langle U_i\rangle_{i\in\mathbb{N}}$ computes the halting problem $\mathbf{0}'$.*

5. Random-freeness and Almost Computability

The counterexample presented in Theorem 5 uses the randomness indexed by pairs, while the counterexample presented in Theorem 7 uses the randomness indexed by $\emptyset$. One might therefore hope to obtain stronger computability results in the context of exchangeable arrays whose representations are constant on these coordinates. In fact, such arrays form a natural class, those whose measures are ergodic and random-free.

An exchangeable array is *ergodic* if and only if it has an Aldous–Hoover representation that does not depend on the first coordinate. An ergodic exchangeable array is said to be **random-free** if it has a representation that further does not depend on the last coordinate, i.e., only depends on the random variables indexed by singletons; such arrays can be described as the random induced substructure of some Borel graph.

Even in this restricted setting, there is not always a computable representation.

Theorem 8 (Avigad, Freer, Roy, and Rute [AFRR16])**.** *There is an ergodic random-free exchangeable array whose distribution is computable, but which has no a.e. continuous representation.*

However, such a distribution does have a representation that satisfies the following standard weaker notion of computability, This notion is sometimes known as L^1-*computability*, and is closely related to *layerwise computability*; see [Miy13, §4] for further details.

Definition 9. A measurable function $f\colon [0,1]^4 \to \{0,1\}$ is **almost computable** if there is a computable sequence of sets $\langle C_n \rangle_{n\in\mathbb{N}}$ and a computable sequence of functions $\langle f_n \rangle_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$,

- each C_n is a closed subset of $[0,1]^4$ with the Lebesgue measure of $[0,1]^4 \setminus C_n$ at most $\frac{1}{2^n}$,
- f_n is computable on C_n, and
- f_n agrees with f on C_n.

No existing programming language supports such a representation for functions. The next result suggests that it may be worthwhile considering programming language constructs that offer the programmer a similar level of flexibility.

Theorem 10 (Avigad, Freer, Roy, and Rute [AFRR16])**.** *Suppose an ergodic random-free exchangeable array has a computable distribution. Then it has an almost computable Aldous–Hoover representation.*

The problem of representing exchangeable structures and the odd shape of potential solutions demonstrate that probabilistic programming raises representational issues that were not obvious in the deterministic setting. It may be necessary to rethink representational assumptions at the foundation of probabilistic programming. Perhaps by providing a convenient interface for "threading through" the desired accuracy of a sample throughout a probabilistic program, we might achieve various representational or efficiency gains.

Acknowledgments

DMR was partially supported by a Newton International Fellowship, Emmanuel Research Fellowship, NSERC Discovery Grant, Connaught Award, and AFOSR grant #FA9550-15-1-0074. This material is based upon work supported by the United States Air Force and the Defense Advanced Research Projects Agency (DARPA) under Contracts No. FA8750-14-C-0001 and FA8750-14-2-0004, the Army Research Office (ARO) grant W911NF-13-1-0212, the Office of Naval Research (ONR) grant N00014-13-1-0333, the National Science Foundation (NSF) grants DMS-0901020 and DMS-0800198, and grants from the John Templeton Foundation and Google. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force, Army, Navy, or DARPA, or the John Templeton Foundation.

References

[AFR16a] N. L. Ackerman, C. E. Freer, and D. M. Roy. *Exchangeable Random Primitives*. Workshop on Probabilistic Programming Semantics (PPS 2016). http://pps2016.soic.indiana.edu/files/2015/12/xrp.pdf. 2016.

[AFR16b] N. L. Ackerman, C. E. Freer, and D. M. Roy. *On the computability of representations of exchangeable structures*. Preprint. 2016.

[AFRR16] J. Avigad, C. Freer, D. Roy, and J. Rute. *On the relative computability of representations of dense graph limits*. Preprint. 2016.

[FR10] C. E. Freer and D. M. Roy. *Posterior distributions are computable from predictive distributions*. In: Proc. 13th Int. Conf. on Artificial Intelligence and Statistics (AISTATS 2010). Vol. 9. J. Machine Learning Research W&CP. 2010.

[FR12] C. E. Freer and D. M. Roy. *Computable de Finetti measures*. Ann. Pure Appl. Logic 163.5 (2012), pp. 530–546.

[GMRBT08] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. *Church: A language for generative models*. In: Proc. of 24th Conf. on Uncertainty in Artificial Intelligence (UAI 2008). Corvalis, Oregon: AUAI Press, 2008, pp. 220–229.

[Kal05] O. Kallenberg. *Probabilistic symmetries and invariance principles*. Probability and its Applications. New York: Springer, 2005.

[Miy13] K. Miyabe. *L^1-computability, layerwise computability and Solovay reducibility*. Computability 2.1 (2013), pp. 15–29.