On the computability and complexity of Bayesian reasoning

Daniel M. Roy
University of Cambridge

Abstract

If we consider the claim made by some cognitive scientists that the mind performs Bayesian reasoning, and if we simultaneously accept the Physical Church-Turing thesis and thus believe that the computational power of the mind is no more than that of a Turing machine, then what limitations are there to the reasoning abilities of the mind?

I give an overview of joint work with **Nathanael Ackerman** (Harvard, Mathematics) and **Cameron Freer** (MIT, CSAIL) that bears on the computability and complexity of Bayesian reasoning. In particular, we prove that conditional probability is in general not computable in the presence of continuous random variables. However, in light of additional structure in the prior distribution, such as the presence of certain types of noise, or of exchangeability, conditioning is possible. These results cover most of statistical practice. At the workshop on Logic and Computational Complexity, we presented results on the computational complexity of conditioning, embedding #P-complete problems in the task of computing conditional probabilities for diffuse continuous random variables. This work complements older work. For example, under cryptographic assumptions, the computational complexity of producing samples and computing probabilities was separated by Ben-David, Chor, Goldreich and Luby. In recent work, we also make use of cryptographic assumptions to show that different representations of exchangeable sequences may have vastly different complexity. However, when faced with an adversary that is computational bounded, these different representations have the same complexity, highlighting the fact that knowledge representation and approximation play a fundamental role in the possibility and plausibility of Bayesian reasoning.

1 Characterizing the boundary of intractability

Over the course of several recent articles [1, 4, 5], we have studied the class of *computable distributions* and *samplable distributions*. This work has been carried out in the formalism of computable analysis, and in particular the type-2 theory of effectivity, which has its roots in Turing's 1936 paper, *On computable numbers, with an application to the Entscheidungsproblem*, and later work by Grzegorczyk and Mazur on computable continuous functions. (See [3] and [15] for introductions.)

Results in this framework pertain to, arguably, the most general setting we might consider, and in particular to a mind computing probabilities for events, or generating samples (hypothesized explanations) from posterior distributions, or, more to the point, making rational or approximately rational decisions in a complex world. In particular, a distribution μ on a metric space is computable exactly when there is a Turing machine[1] that can compute the probabilities for a basis of open balls. Equivalently, because the class of computable and samplable distributions can be shown to align,

[1] Recall that an Turing machine is a finite automaton along with an infinite tape, to which it can write and from which it can read binary digits in the course of its execution. Inputs to the machine are written onto the

μ is computable if and only if there is a probabilistic Turing machine[2] that halts almost surely and whose output distribution is μ.

Given uncertain knowledge represented as a probabilistic program, what prospect is there for updating one's knowledge in light of new observations? In the Bayesian setting, this update is performed by computing a conditional distribution, and so the question is when a mind can update its knowledge about the world if it has the computational power of a Turing machine. We characterize the computability of conditional probability, showing that there are computable joint distributions with noncomputable conditional distributions. The offending object that witnesses this gap is an absolutely continuous distribution on the unit square from which we can generate exact samples. This distribution can be interpreted as a mind reasoning about other minds (Turing machines). In contrast, no version of the conditional distribution of the first axis given the second axis is computable. Such objects are pathological but their existence implies the impossibility of general inference algorithms for the class of computable distributions.

These negative results fly in the face of computational practice and the fact that human minds seem to perform amazing feats of inductive inference. Need we abandon the idea of the mind performing Bayesian reasoning? In short, not necessarily. The continuity of the conditioning variable is essential for proving hardness: conditioning on (computable) discrete random variables is easily seen to be computable using the idea of rejection sampling and the equivalence of computable distributions with samplable distributions. There is also hope for the continuous setting, as one would expect from practice. We prove that the addition of (sufficiently smooth computable and independent) noise to the conditioned random variable renders conditioning computable. More generally, the existence of a conditional density (that is computable *almost everywhere*) of the observed variables given the remaining variables enables conditioning. However, as all the sensory input that our minds process is the product of the unavoidable physical stochasticity of our sensing apparatus, it may be that restricting our attention to either discrete situations or continuous situations with independent noise is ultimately quite appropriate.

Other types of structure also enable conditioning. In the infinite-dimensional setting, e.g., as in Bayesian nonparametrics, when there is often no conditional density, we study exchangeable sequences of random variables and the prospect of posterior inference on the directing random measure. We prove that the posterior distribution of this (potentially infinite dimensional) parameter is computable if and only if the predictive rule[3] is computable. This last result is a straightforward corollary of our work characterizing the computability of de Finetti's well known theorem identifying exchangeable sequences with conditionally i.i.d. sequences. In particular, we prove that an exchangeable sequence is computable if and only if its de Finetti measure (aka mixing measure) is computable.

While computability pertains to the *possibility* of algorithms performing these important operations, complexity pertains to their *plausibility and efficiency*. As claimed above, conditional probabilities are in general not computable. However, calculating conditional probabilities given continuous random variables whose distributions are sufficiently diffuse is shown to be in #P, a class that includes certain forms of integration/counting. More precisely, we show that conditioning polynomially-diffuse random variables is #P-complete, implying that it is the hardest problem, under polynomial-time reduction, in #P. The discrete setting has been studied in the setting of average-case analysis

tape before execution begins, and the output of a Turing machine is the content of its tape when the machine enters its halting state. The details of the underlying finite automata, and the other variations like the number of tapes the machine has access to are relatively inconsequential: there exists a one-tape universal Turing machine which can simulate any other Turing machine, while suffering only a polynomially slowdown.

[2] Probabilistic Turing machines have an additional read-only tape of random bits. It follows that the output of such a machine is itself a random bit string and thus has some distribution. While such a machine seems to only have the ability to represent distributions on bit strings, one can interpret the output of these machines by fixing a surjection from $\{0, 1\}^*$ to a countable space of interest. (This surjection is called a *notation*.) For uncountable spaces, one considers the random output of the probabilistic machine on each input $n \in \mathbb{N}$ (suitably encoded as a binary string on the input tape). This induces a stochastic process on the index set $\mathbb{N}$ whose elements are random bit strings. This space is uncountable and a suitable surjection (called a *representation*) can be used to encode objects like real numbers (e.g., by random Cauchy sequences), functions, distributions, etc. (For more details on this setting, see [15].)

[3] The predictive rule is the conditional distribution of the kth element in the exchangeable sequence given the first $k - 1$ variables, for all k, i.e., $\{P[X_k|X_{1..k-1}]\}_{k\geq 1}$.

and cryptography: e.g., the work of Ben-David, Chor, Goldreich, and Luby [2] implies that, under the assumption that one-way functions exist, the class of efficiently computable distributions is a strict subset of the class of efficiently samplable distributions. This suggests that a mind built to reason by sampling may be able to manipulate a larger space of states of knowledge than one built to compute probabilities. A related argument in favor of sampling over computing probabilities was made by Mansinghka [9], although not from the perspective of this separation.

A relatively unstudied question is the role of computational indistinguishability [6, 16] in evaluating the prospect of a Bayesian mind. Let $X = (X_n)_{n \in \mathbb{N}}$ and $Y = (Y_n)_{n \in \mathbb{N}}$ be sequences of random variables, where each X_n and Y_n are binary strings of length n, and let $\mathcal{A}$ be a set of (potentially random, but independent of X and Y) functions. We say that the process X is $\mathcal{A}$-indistinguishable from the process Y when, for all $A \in \mathcal{A}$, all polynomials p, and sufficiently large n,

$$\Big| \Pr\{A(X_n) = 1\} - \Pr\{A(Y_n) = 1\} \Big| < \frac{1}{p(n)}.$$

The idea is that Y_n may represent an exact sample for some inference task, but we may be willing to accept X_n as "good enough", as measured by the set of tests $\mathcal{A}$. A well-studied special case—that of computational indistinguishability—takes $\mathcal{A}$ to be all (randomized) polynomial time algorithms. Assuming that we are interacting with computational bounded agents, does this give us any leeway to tackle a wider range of inductive inference problems by making gross approximations that are nonetheless undetectable? We believe that it is fruitful to study the computational complexity of generating samples from conditional distributions that are indistinguishable for some class $\mathcal{A}$ (e.g., polynomial-time functions, but potentially other choices depending on the context). It is also possible that we may need to invent new notions of indistinguishability to explain the unreasonable effectiveness of many existing probabilistic models. As one example from recent work, we have returned to the setting of conditional probability in exchangeable sequences and shown that, under cryptographic settings, there are efficiently samplable exchangeable sequences whose posterior predictive distributions are not efficiently samplable. On the other hand, there is an efficiently samplable and computational indistinguishable version of the predictive distribution.

In closing, if the mind performs Bayesian reasoning yet faces the same limitations as Turing machines, then the mind cannot solve all instances of these formally intractable problems of inductive inference. By characterizing this boundary of intractability, we may ultimately shed light on the mind itself.

2 Aside: relationships to machine learning practice

In light of the rising interest in probabilistic programming languages, these theoretical results make contact with the coming generation of machine learning practice. Probabilistic programming languages (e.g., PHA [13], IBAL [12], $\lambda_\circ$[11], Church [7], HANSEI [8], Infer.NET [10], Markov Logic [14], and many more) are an extreme end point of the search for compact representations for specifying probabilistic models. Many of these languages adopt modern programming language syntax and thus can handle modeling idioms (like recursion, and in many cases, higher-order functions) that are difficult to represent with graphical models. There is a substantial effort being invested into designing general purpose algorithms for computing posterior probabilities or samples using these representations, and this suggests that there is a need for a clear understanding of the mathematical limitations of automating probabilistic inference. The limitative results described above imply that no general algorithm exists for conditioning and that we must instead decide to support a particular collection of special cases that hopefully cover the space of important (or in the case of the mind, natural) problems of inductive inference. Which ones we choose remains an interesting, largely ignored question.

The rising interest in probabilistic programming mirrors the situation with flexible Bayesian models, and especially nonparametric ones built from stochastic processes. In this setting one is interested in placing prior distributions on large spaces like function classes. Representational choices becomes very important in these settings, and the theory of computable metric spaces, computable topological spaces, etc., can offer some guidance. Moreover, a computational viewpoint provides a perspective on what one might consider the minimum reasonable requirement of a representation: the likelihood should be an (efficiently) computable function.

Acknowledgments

DMR is supported by a Newton International Fellowship and Emmanuel College. The author would like to thank Nate Ackerman, Cameron Freer and Vikash Mansinghka for comments on drafts.

References

[1] N. L. Ackerman, C. E. Freer, and D. M. Roy. Noncomputable conditional distributions. In *Proc. Logic in Computer Science*, 2010. URL http://danroy.org/papers/AckFreRoy-LICS-2011.pdf.

[2] 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. ISSN 0022-0000. doi: 10.1016/0022-0000(92)90019-F. URL http://dx.doi.org/10.1016/0022-0000(92)90019-F.

[3] M. Braverman and S. Cook. Computing over the reals: foundations for scientific computing. *Notices Amer. Math. Soc.*, 53(3):318–329, 2006. URL http://www.ams.org/notices/200603/fea-cook.pdf.

[4] 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. URL http://jmlr.csail.mit.edu/proceedings/papers/v9/freer10a/freer-roy10a.pdf.

[5] C. E. Freer and D. M. Roy. Computable de Finetti measures. *Ann. Pure Appl. Logic*, 2011. ISSN 0168-0072. doi: 10.1016/j.apal.2011.06.011. URL http://dx.doi.org/10.1016/j.apal.2011.06.011.

[6] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. URL http://dx.doi.org/10.1016/0022-0000(84)90070-9.

[7] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: a language for generative models. In *Proc. of the 24th Conf. on Uncertainty in Artificial Intelligence*, 2008. URL http://danroy.org/papers/church_GooManRoyBonTen-UAI-2008.pdf.

[8] 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. URL http://dx.doi.org/10.1007/978-3-642-03034-5_17.

[9] V. K. Mansinghka. *Natively Probabilistic Computing*. PhD thesis, Massachusetts Institute of Technology, 2009. URL http://web.mit.edu/vkm/www/vkm-dissertation.pdf.

[10] T. Minka, J. Winn, J. Guiver, and D. Knowles. Infer.NET 2.4, 2010. URL http://research.microsoft.com/infernet. Microsoft Research Cambridge.

[11] S. Park, F. Pfenning, and S. Thrun. A probabilistic language based on sampling functions. *ACM Trans. Program. Lang. Syst.*, 31(1):1–46, 2008. URL http://dx.doi.org/10.1145/1452044.1452048.

[12] 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. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1299.

[13] D. Poole. Representing Bayesian networks within probabilistic Horn abduction. In *Proc. of the 7th Conf. on Uncertainty in Artificial Intelligence*, pages 271–278, 1991. URL http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.5481.

[14] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006. URL http://dx.doi.org/10.1007/s10994-006-5833-1.

[15] K. Weihrauch. *Computable analysis: an introduction*. Springer-Verlag, Berlin, 2000. URL http://www.springer.com/computer/theoretical+computer+science/book/978-3-540-66817-6.

[16] A. C.-C. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91. IEEE Computer Society, 1982. URL http://doi.ieeecomputersociety.org/10.1109/SFCS.1982.45.