When are probabilistic programs probably computationally tractable?

Cameron E. Freer
Univ. of Hawai'i at Mānoa
freer@math.hawaii.edu

Vikash K. Mansinghka
Navia Systems
vkm@naviasystems.com

Daniel M. Roy
MIT CSAIL
droy@csail.mit.edu

Abstract

We study the computational complexity of Bayesian inference through the lens of simulating probabilistic programs. Our approach is grounded in new conceptual hypotheses about some intrinsic drivers of computational complexity, drawn from the experience of computational Bayesian statisticians. We sketch a research program to address two issues, via a combination of theory and experiments:

(a) What conditions suffice for Bayesian inference by posterior simulation to be computationally efficient?

(b) How should probabilistic programmers write their probabilistic programs so as to maximize their probable tractability?

The research program we articulate, if successful, may help to explain the gap between the unreasonable effectiveness of some simple, widely-used sampling algorithms and the classical study of reductions to Bayes from hard problems of deduction, arithmetic, and optimization.

Predicting the runtime of programs in knowledge-based systems, where executing a program involves applying a general-purpose reasoning algorithm to some knowledge representation, has always been challenging. In logic programming, programmers were taught to guess the tractability of their programs by acquiring intuition from direct experience implementing complex general reasoning algorithms as well as a large body of special cases. The practice of computational Bayesian statistics is quite similar: even when using general purpose tools like BUGS [LTBS00], experts rely on intuitions derived from lengthy experience implementing specialized samplers for specific model classes [GRS96].

If probabilistic programming (and computational Bayesian statistics, more generally) is to spread beyond the small community of Bayesian statisticians and inference engine designers, we need to dramatically improve this learning curve. In particular, we need to empower domain experts to build and perform inference over rich, realistic models without an extended education in Monte Carlo. In service of the eventual goal of producing a simple, teachable methodology, we focus on the following two tasks:

(a) Identify broadly applicable sufficient conditions under which probabilistic programs (including, but not limited to, Bayesian inference problems represented as probabilistic programs) are computationally tractable.

(b) Develop a clear, communicable design methodology for probabilistic programs, combining these sufficient conditions and intuitions derived from experiments, that lets probabilistic programmers solve problems in terms of programs that will probably terminate quickly enough and with acceptable accuracy. At minimum, this methodology should help probabilistic programmers detect and resolve situations where it seems unlikely that rapid, accurate termination is feasible.

We have been attempting to address these issues, and here propose a research program that bridges the gap between the very different intuitions from deterministic computational complexity theory and the experience of computational Bayesian statisticians. The sketch we present here includes new theory and experiments as well as a reinterpretation of some recently developed theory. In this note, we present three claims, along with some of the theoretical and empirical evidence we have gathered so far that lends support to each. We briefly collect the claims here:

1. Surprise upper-bounds the complexity of exact posterior simulation. In fact, the surprise in the data can sometimes provide a sharper characterization of the computational tractability of posterior simulation than measures based on, e.g., the dimensionality, treewidth of an underlying graphical model, etc.
2. Approximate posterior simulation is sometimes quite tractable, especially when either the surprise can be broken into small steps, or the constraints (induced by the data) on the program's random choices are sparsely overlapping.
3. Probabilistic programmers can increase the likely tractability of their programs, without resorting to tempering methods, by designing them to be self-relaxing.

Throughout, our perspective is informed by the development and use of probabilistic programming systems that have two unusual properties. First, they perform inference by stochastic simulation, via either universal rejection or universal MCMC. Models are represented by programs that generate samples, and inference algorithms use this source code (sometimes by running fragments of it) to produce another stochastic process that simulates from the posterior. This introduces a tight coupling between the properties of the program representing the prior and the stochastic process that carries out inference. For example, the time and space complexity of the prior program affects the optimal time and space complexities of each inference iteration. Additionally, the way that entropy flows from the prior to the posterior becomes more clearly related to the complexity of inference.

Second, these probabilistic programming systems make it natural to construct random objects that are as rich as those definable in deterministic functional programming languages. For example, in Church [GMR+08] and in Monte (a Church-like language developed at Navia), one can have variables representing random data structures and procedures, and recursive processes whose termination criteria depend on random choices. This expressiveness makes it easy to introduce arbitrarily structured randomness into a probabilistic program.

This is very different from the experience of those Bayesian statisticians who hand-design new variational or MCMC algorithms each time the underlying model changes, and also from systems like BUGS [LTBS00] and Infer.NET [MWGK10], which make different choices with respect to model expressiveness. The combination of these differences has led us to very different intuitions about the intrinsic hardness of various inference problems and the shape of strategies that might help to improve the tractability of inference.

1. Surprise upper-bounds the complexity of exact posterior simulation.

We have two constructive results showing that exact posterior simulation via rejection is tractable when the data are not too surprising, or when the surprise can be broken down into a short enough sequence of sufficiently unsurprising steps. These results provide a conservative estimate of the difficulty of practical probabilistic programming. They also sharpen those classical worst-case perspectives on exact inference that are based on syntactic criteria such as the bare dimensionality of the support of the distribution, treewidth of a graphical model, etc. Importantly, our results do not underestimate the anticipated complexity of problems believed to be truly hard (e.g., inverting cryptosystems).

Let `<exp>` be an expression in the probabilistic programming language Church [GMR+08] describing a stochastic process that induces the prior distribution $p(x)$ of some Bayesian reasoner. To sample from that prior in Church, one can apply the procedure `eval` to it, for example by evaluating `(eval <exp>)`. Let `<pred>` be a predicate of one formal argument X (a sample from `<exp>`) that produces either true or false. Let Y be the binary random variable corresponding to the output of the predicate. Together, `<exp>` and `<pred>` induce a joint distribution $p(x, y)$ on the pair of random variables (X, Y), where X can be an arbitrary Church value (e.g., a number, list, string, data structure, procedure, etc.)

In Church, applying the `query` procedure to `<exp>` and `<pred>` (e.g., by evaluating the expression `(query <exp> <pred>)`) results in a sample from the conditional distribution $p(x|y = 1)$ of X given than $Y = 1$. One way this can be used to encode Bayesian inference is by letting X be a tuple (H, D) of some hypothesis and some data sampled from the prior, and letting the predicate check that D is equal to the actual data that was observed.

Proposition. *Let N be the number of attempts before a rejection sampler for* `(query <exp> <pred>)` *succeeds when using samples from the distribution induced by* `<exp>` *and* `<pred>`*. Assume the application of* `<pred>` *to any input consumes no randomess, i.e.,* `<pred>` *is deterministic. Then N is geometrically distributed with mean* $\exp(D_{KL}(\texttt{(query <exp> <pred>)} \| \texttt{(eval <exp>)}))$.

Proof. The probability that a sample from $p(x, y)$ satisfies $y = 1$ is precisely $p(y = 1)$. Therefore the number of attempts before a sample is accepted is geometrically distributed with expectation $1/p(y = 1)$. Let $\mathbf{E}_{x|y=1}$ denote expectation with respect to the posterior distribution $p(x \mid y = 1)$. We have

$$D_{\mathrm{KL}}(p(x \mid y = 1) \,\|\, p(x)) = \mathbf{E}_{x|y=1} \log \tfrac{p(x \mid y=1)}{p(x)} \tag{1}$$

$$= \mathbf{E}_{x|y=1} \log \frac{p(x, y = 1)}{p(x)\, p(y = 1)} \tag{2}$$

$$= \log \frac{1}{p(y = 1)} + \mathbf{E}_{x|y=1} \log p(y = 1 \mid x). \tag{3}$$

The random variable Y is derived from X by a deterministic function, and so $p(y = 1 \mid x) = 1$ for all x in the support of $p(x \mid y = 1)$. Hence the second term of (3) is zero. It follows immediately that

$$\frac{1}{p(y = 1)} = e^{D_{\mathrm{KL}}(p(x \mid y=1) \,\|\, p(x))}. \tag{4}$$

□

If $D_{\mathrm{KL}}(p(x \mid y = 1) \,\|\, p(x)) = O(\log n)$, then the expected number of rejections is polynomial in n. If in addition <exp> can be efficiently evaluated (i.e., $p(x)$ is P-samplable [BCGL92; Yam99; Aar10]) and <pred> is efficiently computable, then rejection sampling is tractable (and the posterior is also P-samplable). The bridge this proposition provides to the computational complexity of inference by posterior simulation stems from probabilistic programming, where priors and posteriors are *represented* in terms of programs that sample from them, and therefore come equipped with time and space complexities. This result is related to folklore in cryptography [TTV09] and results in communication complexity [HJMR07; HJMR10].

Figure 1 explores posterior simulation by rejection in a simple but illustrative model. While we do not expect monothlithic rejection sampling to be an adequate inference strategy in general, we think it points in an interesting direction for three reasons. First, it provides an upper bound on the computational tractability of Bayesian inference that depends on *semantic* properties of the distributions involved. Contrast this to the syntactic criteria emphasized by traditional analyses of the complexity of MAP and marginal inference, such as dimensionality, treewidth, and so on. Second, we have sometimes found ourselves surprised by the effectiveness of rejection on problems with a large (or even infinite) number of random variables and a high degree of noise, but only a handful of data points. Our result helps to explain why rejection might actually be a useful strategy in those settings. Finally, we believe that the result suggests that there are opportunities for new applications of rejection sampling in computational Bayesian statistics, especially in simulation from the low-dimensional conditional distributions that arise in MCMC algorithms such as Gibbs sampling. Typical Gibbs algorithms use numerical techniques to simulate from the conditional posterior over one or several variables. However, if that posterior is nearly uniform or only moderately peaky, or not too far from some easy-to-update conditional proposal (potentially obtained by quite heuristic means), our result suggests that rejection sampling might actually be faster than standard numerical approaches for Gibbs sampling. For example, in blocked Gibbs sampling several cluster assignments in a Dirichlet process mixture, it might be that the relative entropy from the conditional posterior on the block of assignments to the prior is low enough (e.g., due to noise in the observations) to make rejection more favorable than discrete numerical blocked Gibbs.

Finally, the Kullback–Leibler divergence (D_{KL}) governing the complexity of rejection, taken from the posterior to the prior, also determines the sample complexity bounds from PAC Bayes [McA03]. Sloganized, then, this result says that concept classes that admit tight generalization bounds also admit tractable rejection samplers, and vice versa.

Proposition. *The probability that sequential rejection sampling* [MRJT09] *succeeds is exponential in the sum of the KL divergences between each successive distribution in the sequence of restrictions.*

The idea behind sequential rejection is to convert a monolithic posterior simulation problem into a nested sequence of easier posterior simulation problems, where simulating successfully from each provides the prior for rejection sampling for the next. Of course such a sequence may be hard to find or may not exist. For concreteness, consider the case of simulating from the joint distribution implied by a discrete variable factor graph. This can be written as posterior simulation by an auxiliary variable scheme: use a uniform prior over the variables, with one additional boolean random variable for each factor, which takes the value true with probability exponential in the (negative) energy of that factor. Simulating from the joint then

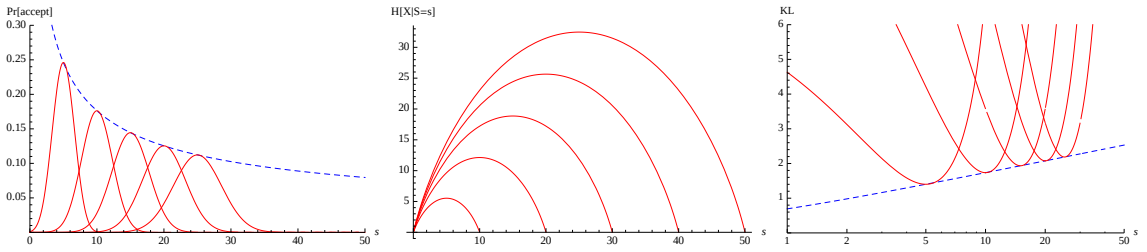

Figure 1: The following example illustrates that rejection sampling doesn't always scale poorly with the dimension of the problem. Let $\{X_i\}_{i\geq 1}$ be independent Bernoulli($\frac{1}{2}$)-distributed random variables. For each n, define $S_n = \sum_{i=1}^{n} X_i$ and consider the prior distribution $\mathbf{P}_n$ on $X_{1:n} = (X_1, \ldots, X_n)$ and, for each $s \in \{0, 1, \ldots, n\}$, the posterior distribution $\mathbf{P}_{n|s}$ on $X_{1:n}$ given $S_n = s$. **(left)** The probability that a sample from $\mathbf{P}_n$ satisfies $S_n = s$ (and is thus a sample from the posterior $\mathbf{P}_{n|s}$) is simply the marginal probability that $\mathbf{P}(S_n = s)$, which follows a Binomial(n, $\frac{1}{2}$) distribution. In this figure we have plotted the probability mass function for the Binomial(n, $\frac{1}{2}$) distribution for the cases $n = 10, 20, 30, 40, 50$. For each size n and observation s, the plot indicates the probability that a sample from $\mathbf{P}_n$ is accepted as a sample from $\mathbf{P}_{n|s}$. The *blue dashed* line passes through the peak of each probability mass function where $s = \frac{n}{2}$, tracing the probability of accepting a sample from $\mathbf{P}_n$ for $\mathbf{P}_{n|\frac{n}{2}}$. This function is given by the expression $2^{-2s}\binom{2s}{s} \approx (s\pi)^{-1/2}$, and evidently decays at a polynomial rate. The same rate of decay is achieved asymptotically for the point $\frac{n}{2} - c$ for a constant c. **(center)** The posterior $\mathbf{P}_{n|s}$ is uniformly distributed on all length-n bit sequences with s ones. Therefore the posterior entropy is $\log\binom{n}{s}$ and is thus maximized at $s = n/2$ with the approximate value $n \log 2 - \frac{1}{2}\log n$, while the prior entropy is $n \log 2$. Note that both grow asymptotically at the same rate. **(right)** While the rate of decay of the acceptance probability may be hard to judge from the leftmost plot, the right plot, which is a log-linear plot of the KL divergence $D_{\mathrm{KL}}(\mathbf{P}_{n|s} \,\|\, \mathbf{P}_n)$, provides a clear depiction of the polynomial rate. The minima $D_{\mathrm{KL}}(\mathbf{P}_{n|\frac{n}{2}} \,\|\, \mathbf{P}_n)$ follow a line, and thus the acceptance rate $\exp(-D_{\mathrm{KL}})$ of rejection sampling with this data is decaying polynomially. While this binomial example possesses additional structure (analyzable via the central limit theorem) that is responsible for the effectiveness of rejection, we suggest that noisy generative processes (or some of their low-dimensional conditional distributions) might be far more amenable to inference by rejection than one might naively guess.

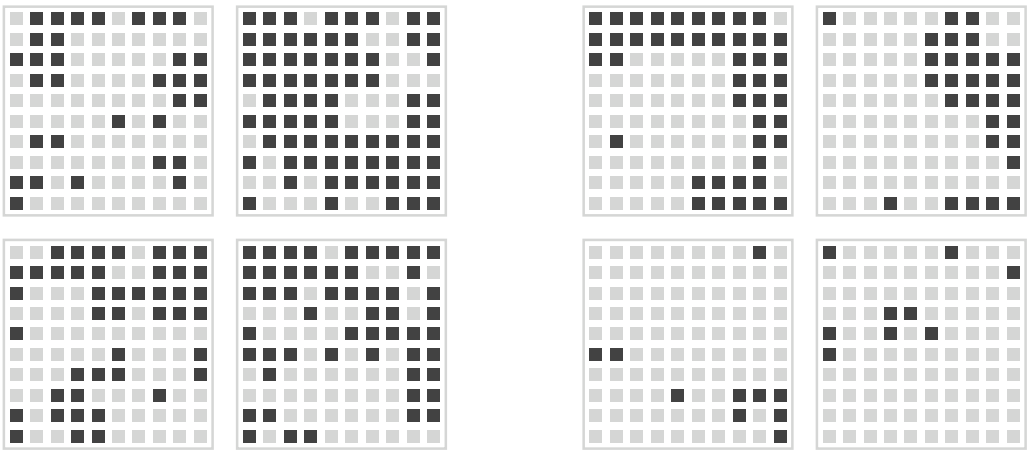

Figure 2: (left 4) Exact samples from a 10x10-dimensional grid ferromagnetic Ising just below the critical temperature. **(right 4)** Exact samples from a 10x10-dimensional grid ferromagnetic Ising just above the critical temperature. All of these samples were obtained by sequential rejection, in a setting where theory and intution for monolithic rejection might lead one to expect exact simulation to be infeasible (without advanced Markov chain coupling techniques). (Source: [MRJT09])

reduces to rejection sampling subject to the constraint that all the factor-variables are true. Sequential rejection amounts to successively simulating from subgraphs of the original factor graph; for more details, see [MRJT09].

One might expect any rejection-based strategy to fail catastrophically as the number of variables grows. For example, while the uniform prior on spins in a highly coupled, N by N ferromagnetic Ising has N^2 bits of entropy, the joint distribution has roughly 1 bit. Our monolithic rejection result then shows that the probability of acceptance will be very low, as the relative entropy gap is too large to be closed. However,

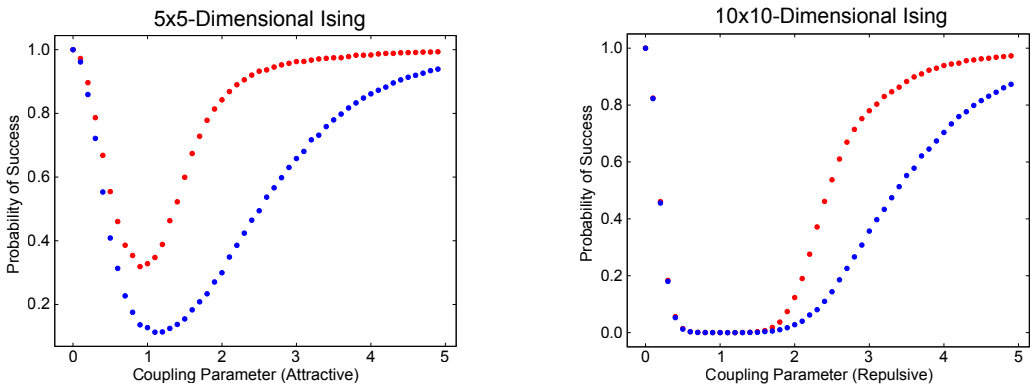

Figure 3: (left) Ferromagnetic. **(right)** Antiferromagnetic. **(both)** Frequency of acceptance in nonadaptive (blue, lower) and adaptive (red, higher) sequential rejection as a function of coupling strength J. Monolithic rejection approaches suffer from exponential decay in acceptance probability with dimension across all coupling strengths, while generic MCMC approaches like Gibbs sampling fail to converge (even approximately) when the coupling reaches or exceeds the critical value. In contrast, sequential rejection can tractably generate samples in the very low and very high coupling regimes, where the surprise introduced by the constraints is either very low or can be broken down sequentially into a much smaller total amount. Both non-adaptive and adaptive sequential rejection struggle near the critical region, though adaptation yields significant benefits (that cannot yet be analyzed by the theory we have been developing). (Source: [MRJT09])

sequential rejection succeeds in these settings by breaking the surprise down into bite-sized pieces whose KL divergences sum to a number that is much smaller than the KL between the posterior and prior. A good sequential rejection sequence consumes most of the posterior entropy early on in the sequence. In the case of the highly coupled ferromagnetic Ising, sequential rejection succeeds by first choosing which of the two modes to be in (by setting the spin value of one site), then conditioning on that choice to complete rejection sampling on the rest of the lattice. As the high coupling strength strongly constrains each of those subsequent choices, very little entropy remains, and the overall acceptance probability can be surprisingly high.

2. Approximate posterior simulation is sometimes quite tractable, especially when either the surprise can be broken into small steps, or the constraints (induced by the data) on the program's random choices are sparsely overlapping.

The theory and experiments supporting our first claim suggest that Bayesian inference by exact posterior simulation may be surprisingly tractable, when either the surprise in the data is low or it can be broken down into a sequence that is less surprising overall. However, we do not expect the majority of probabilistic programs to be tractable in this way, and we have a large body of experience with approximate posterior simulation techniques (e.g., MCMC, SMC, and hybrids) that often seem to work quite well even when the overall level of surprise in the data is quite great. We do not know the contours of the theory and experiments that will let us understand sufficient conditions where approximate posterior simulation will be tractable. However, we can provide two pointers that approach this issue in quite a different way than the traditional eigenvalue-gap analyses of MCMC convergence that have been classically motivated by the desire to bound the variance of Monte Carlo estimates.

First, for approximate sampling based on tempered transitions [Nea96], recent work [BFH10] demonstrates that the optimal sequence of temperatures is obtained by minimizing the sum of the symmetrized KL divergences between successive distributions in the ladder of tempering distributions. While it may in general be as hard (or harder) to compute the optimal ladder as to solve the original problem[1], we think this connection is interesting in part because it links semantic quantities similar to the ones we consider for exact simulation to the tractability of annealing-type methods, in a way that helps to remove some of the arbitrary degrees of freedom that have plagued practitioners trying to apply those methods.

[1] In fact, even computing the symmetrized KL divergence between two proposed rungs in a ladder may be hard. Compare this with the fact that choosing the optimal ordering for variable elimination is intractable, yet it is tractable to compute the runtime for a given elimination ordering.

Second, recent work [HSS10] analyzing the constructive proof [MT10] of the Lovász Local Lemma (LLL) [EL75] has shown that a very simple randomized algorithm, based on Markov chain iteration (though with an unusual termination criterion), can approximately implement a range of combinatorially-oriented probabilistic programs with provably fast convergence, in a rather general setting. This work produces a bridge between the classical uses of the LLL (as a probabilistic tool to prove the existence of combinatorial objects) and approximate posterior simulation.

Proposition (LLL [EL75]). *Suppose each of the events $A_1, \ldots, A_m$ is independent of all but d of the others, and each has marginal probability $\leq p$. If $p(d+1) \leq e^{-1}$ then there is positive probability that no event occurs.*

Algorithm 1 MT($P_1, \ldots, P_n; A_1, \ldots, A_m$) [MT10]

Require: LLL conditions on the probabilities of P_i and overlap of A_j.
Ensure: An assignment of $P_1, \ldots, P_n$ for which none of $A_1, \ldots, A_m$ hold.

for all P_i **do**
    independently assign P_i at random (according to respective distribution)
while there is a violated event **do**
    pick an arbitrary violated event A_j
    for all $P_i \in A_j$ **do**
        independently assign P_i at random (according to distribution of P_i)

Algorithm 2 MT-SAT(φ) [MT10]

Require: $\varphi = C_1 \wedge \cdots \wedge C_m$ is a k-SAT formula in n variables $v_1, \ldots, v_n$ such that every variable appears in at most $2^k/(ek)$ clauses.
Ensure: An assignment of variables $v_1, \ldots, v_n$ for which φ is satisfied.

for all v_i **do**
    independently assign v_i uniformly at random
while some clause is unsatisfied **do**
    pick an arbitrary unsatisfied event C_j
    for all $v_i \in C_j$ **do**
        independently assign v_i uniformly at random

Proposition ([HSS10]). *Suppose $P_1, \ldots, P_n$ are mutually independent random variables and $A_1, \ldots, A_m$ are events each determined by some subset of the variables. Under the same sufficient conditions as in the above proposition efficiently constructs an assignment of the variables P_i for which no event A_j holds. In particular, in most known applications, the algorithm terminates in an expected $O(n^2 \log n)$ number of resamplings.*

The algorithm MT-SAT constitutes a special case of MT, induced by an embedding of a k-SAT problem (with each A_j being the event that some particular clause does not hold) as an instance of the LLL.

Furthermore, [HSS10] give an analysis of the distribution D' on variable assignments induced by Algorithm 1, and show a sense in which it approximately samples from the conditional distribution D of the variables P_i given that no event A_j holds. In particular, D and D' induce similar probabilities on any event that does not overlap with too many events A_j. While only an unusual subset of probabilistic programs is directly covered by these conditions, the analytical methodology and underlying intuitions suggest a very new approach to the problem of MCMC convergence.

3. Probabilistic programmers can increase the likely tractability of their programs, without resorting to tempering methods, by designing them to be self-relaxing.

In addition to the sufficient conditions we have outlined above, we offer a step towards a design discipline that probabilistic programmers can follow which, so far, seems to increase the probability that a stochastic simulation based implementation (e.g., the universal MCMC algorithms in Church [GMR+08]) will be able to tractably answer a wide range of queries. The key idea takes inspiration from an aspect of evolution: complex structures often arise in part because there is a smooth bridge along which fitness mostly increases, running through intermediate structures. The very high dimensional nature of the space of possible structures provides the degrees of freedom that enable these smooth paths to exist. Annealing-type methods attempt to introduce this kind of structure in a post-hoc fashion, by adding variables like temperature (or by co-opting variables like various weighting parameters) and smoothly changing them in ways that facilitate mixing.

Our approach is motivated by a different intuition: one driver of slow convergence (supported by both our experience and the sufficient conditions we previously mentioned) is strong global dependence. When large groups of variables are constrained in a near-deterministic fashion, rejection often becomes unlikely to succeed, and generic inference techniques based on modifying a small number of variables one at a time often converge slowly. We leverage the fact that probabilistic programming systems make it very easy to

account for real, but usually overlooked, uncertainty (e.g., by replacing constants with random variables) and to relax artificial determinism in some model into a more realistic stochastic process. For example, in a generative model for ranking game players based on skill levels [DHMG07], one can replace deterministic numerical comparisons with stochastic comparators whose distribution depends on the degree to which the two quantities are different.

This capability, which might normally be viewed as decreasing tractability (e.g., because it adds dimensionality or increases treewidth), appears to have the opposite effect in many important situations. In particular, by admitting our uncertainty about things we are used to assuming constant (or the stochasticity of elements of our processes that we are used to modeling deterministically), we have often found that generic simulation-based inference techniques perform better. For example, universal MCMC or naive single-site Gibbs for Dirichlet process mixtures of binomials often seem to mix faster when uncertainty about the Dirichlet process and component prior hyperparameters is incorporated: the move between a solution with all datapoints in a single cluster and a solution with each datapoint in its own cluster is smoothest when the hyperparameters favor cleaner clusters and additional groups. As another example, polynomial curve-fitting (see Figure 4) can be made self-relaxing by incorporating uncertainty about the amount of noise in the data (in both x and y). We have anecdotally experienced sufficient success with this approach — as an epistemologically-motivated alternative to tempering — to offer it as a design guideline to increase the probability of ending up with tractable probabilistic programs. The hard work of determining the merits of this approach through rigorous empirical and theoretical investigation remains to be done.

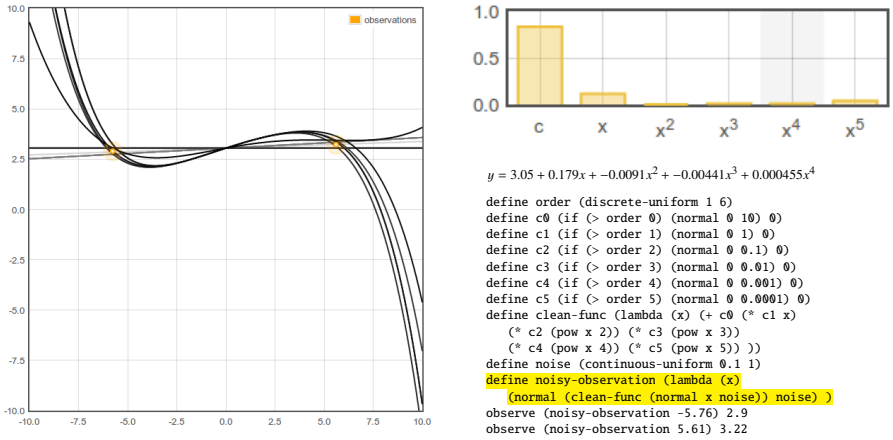

Figure 4: Degree 5 polynomial curve-fitting to 2 observed data points. **(left)** Posterior samples given observed data. **(top right)** Bar plot illustrating the conditional distribution over the degree of polynomial approximation, given the observations. **(bottom right)** Program listing in the Monte language (inspired by but different from Church) from Navia Systems. Mode switching occurs via high noise levels, where it is possible to move more freely between polynomials of different degrees, even if this involves passing through worse-fitting polynomials of intermediate degree. For example, the inference engine can first increase the expected noise level and then move from a well-fitting constant to a less well-fitting quadratic and then decrease the noise once more reasonable coefficients have been inferred. Precisely documenting the dynamics of these effects of self-relaxing programs and quantifying their impact on convergence remains an important project for future work.

Intuitively, a probabilistic program is *self-relaxing* if between all pairs of hypotheses, there is a short path with smoothly varying joint probability composed of steps traversed by small changes to a small number of random choices. Self-relaxing programs mesh nicely with posterior simulation via, e.g., MCMC: the sequence of choices provides trajectories that connect what might otherwise be widely separated islands of high probability. One approach to probabilistic program design that may encourage tractability, then, is to encourage probabilistic programmers to produce self-relaxing programs where possible, because then we believe that a variety of simulation-based inference techniques are likely to perform well. One especially intriguing way to achieve self-relaxation is to examine all fixed constants and deterministic subprocesses and replace them with at least one layer of stochastic choice. We are interested in exploring the degree to which we can rely on generic sampling to both utilize those degrees of freedom to improve mixing and to ultimately settle (in the posterior) on reasonable values.

We anticipate that some problems, such as inverting cryptosystems, accurately solving very ill-conditioned (but nonsingular) linear systems, or SAT solving in the hard phase [GS05] are intrinsically computationally hard. So far, self-relaxation seems consistent with these cases: while these problems can be relaxed into easy ones in a variety of ways, these relaxations bear little resemblance (e.g., in terms of KL) to the original problem.

Example. *Solving a n-dimensional linear system $A\bar{x} = \bar{b}$ for $\bar{x}$ can be embedded as a limit of posterior simulation problems, where $\bar{x} \sim \mathcal{N}(0, \epsilon_x I^{(n)})$ and $\bar{b}|\bar{x} \sim \mathcal{N}(A\bar{x}, \epsilon_y I^{(n)})$. Let $\rho = \epsilon_y/\epsilon_x$. Then the posterior distribution of $\bar{x}$ given $\bar{y}$ is normally distributed with mean*

$$\mu = \left(A'A + \rho I^{(n)}\right)^{-1} A'\bar{b} \tag{5}$$

and covariance

$$\Sigma = \epsilon_y \left(A'A + \rho I^{(n)}\right)^{-1}. \tag{6}$$

Assume that A is invertible. As $\epsilon_y \to 0$, we have that $\rho \to 0$, hence $\Sigma \to 0$ and $\mu \to A^{-1}\bar{b}$.

When A is ill-conditioned, it is widely believed that accurately solving for $\bar{x}$ is intrinsically hard (e.g., iterative methods converge slowly, and some algebraic methods may require unreasonable numerical precision). This connects with our understanding of self-relaxation as follows:

Assume we observe that $\bar{b} = 0$. Then the KL divergence between an unrelaxed posterior ($\rho = 0$) and relaxed posterior is

$$\frac{1}{2}\sum_{i=1}^{n}\left(\log\frac{\sigma_i^2+\rho}{\sigma_i^2} + \frac{\sigma_i^2+\rho}{\sigma_i^2}\right) - \frac{n}{2}, \tag{7}$$

where σ_i are the singular values of A. Assume, without loss of generality, that $\sigma_i > \sigma_{i+1}$. Then $\kappa = \frac{\sigma_1}{\sigma_n}$ is the condition number of A. In order to reduce the condition number of $A'A$ (which appears in (6)) by a factor of γ, we must take ρ to be

$$\frac{\sigma_1\sigma_n(\gamma-1)}{\sigma_1 - \sigma_n\gamma}. \tag{8}$$

However, this increase in ρ causes the KL divergence to increase by at least $(1 - 1/\kappa)\,\gamma$. If $\kappa \gg 1$, i.e., A is ill-conditioned, then relaxing the system by adding independent noise rapidly washes out the underlying problem, resulting in a new posterior quite close to the prior but very far from the true posterior. In the extreme case where A is singular, the dimensionality of the support of the posterior changes abruptly when the noise level is greater than 0.

Discussion

We believe that the real challenge underlying these issues is the development of a theory of natively probabilistic computational complexity [Man09], centered on the *computational difficulty of stochastic simulation, especially posterior simulation, as an end unto itself*, rather than as a preliminary for Monte Carlo function approximation. The structure of such a theory could arise from considerations such as the rate at which surprise arises as a function of data, or the amenability of different program classes to self-relaxation. Typical problems would involve conditioning noisy prior processes, with traditional deterministic problems of optimization, counting, and satisfiability arising as deterministic limits.

Many researchers have eschewed Bayesian reasoning due to its apparent computational intractability — often due to the dire picture presented by classical theory, though sometimes also due to difficulties in practice. We wonder if, instead, the apparent intractability of Bayes might be a sign that our perspective on computational complexity might benefit from reconsideration.

Acknowledgements

The authors would like to thank Isaac Chuang, Jonathan Kelner, John Langford, Ross Lippert, Manfred Opper, Stuart Russell, and Josh Tenenbaum for helpful discussions, Ruslan Salakhutdinov and David Wingate for comments on a draft, and Keith Bonawitz for his work on the curve-fitting example. C.E.F. is partially supported by NSF grant DMS-0901020.

References

[Aar10] S. Aaronson, *The equivalence of sampling and searching*, arXiv:1009.5104 (2010).

[BCGL92] S. Ben-David, B. Chor, O. Goldreich, and M. Luby, *On the theory of average case complexity*, J. Comput. System Sci. **44** (1992), no. 2, 193–219.

[BFH10] G. Behrens, N. Friel, and M. Hurn, *Tuning tempered transitions*, arXiv:1010.0842 (2010).

[DHMG07] P. Dangauthier, R. Herbrich, T. Minka, and T. Graepel, *TrueSkill through time: Revisiting the history of chess*, Advances in Neural Information Processing Systems 20, 2007, pp. 337–344.

[EL75] P. Erdős and L. Lovász, *Problems and results on* 3*-chromatic hypergraphs and some related questions*, Infinite and finite sets (Colloq., Keszthely, 1973), Vol. II, North-Holland, Amsterdam, 1975, pp. 609–627.

[GMR+08] N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum, *Church: a language for generative models*, Uncertainty in Artificial Intelligence, 2008.

[GRS96] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in practice*, Chapman and Hall, 1996.

[GS05] C. Gomes and B. Selman, *Can get satisfaction*, Nature **435** (2005), no. 7043, 751–752.

[HJMR07] P. Harsha, R. Jain, D. McAllester, and J. Radhakrishnan, *The communication complexity of correlation*, IEEE Conference on Computational Complexity, IEEE Computer Society, 2007, pp. 10–23.

[HJMR10] _____, *The communication complexity of correlation*, IEEE Trans. Inf. Theor. **56** (2010), no. 1, 438–449.

[HSS10] B. Haeupler, B. Saha, and A. Srinivasan, *New constructive aspects of the Lovász local lemma*, 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE Computer Society, 2010.

[LTBS00] D. J. Lunn, A. Thomas, N. Best, and D. Spiegelhalter, *WinBUGS - a Bayesian modelling framework: concepts, structure, and extensibility*, Statistics and Computing **10** (2000), 325–337.

[Man09] V. K. Mansinghka, *Natively probabilistic computation*, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[McA03] D. McAllester, *Simplified PAC-Bayesian Margin Bounds*, Learning theory and Kernel machines: COLT/Kernel 2003, Springer Verlag, 2003, p. 203.

[MRJT09] V. K. Mansinghka, D. M. Roy, E. Jonas, and J. B. Tenenbaum, *Exact and approximate sampling by systematic stochastic search*, AISTATS 2009 (2009).

[MT10] R. A. Moser and G. Tardos, *A constructive proof of the general Lovász local lemma*, J. ACM **57** (2010), no. 2, 1–15.

[MWGK10] T. Minka, J. Winn, J. Guiver, and D. Knowles, *Infer.NET 2.4*, 2010, Microsoft Research Cambridge. http://research.microsoft.com/infernet.

[Nea96] R. M. Neal, *Sampling from multimodal distributions using tempered transitions*, Statistics and Computing **6** (1996), 353–366.

[TTV09] L. Trevisan, M. Tulsiani, and S. P. Vadhan, *Regularity, boosting, and efficiently simulating every high-entropy distribution*, IEEE Conference on Computational Complexity, IEEE Computer Society, 2009, pp. 126–136.

[Yam99] T. Yamakami, *Polynomial time samplable distributions*, J. Complexity **15** (1999), no. 4, 557–574.