Introduction

- Markov chain Monte Carlo (MCMC) algorithms are workhorses for approximate inference in probabilistic programs.
- MCMC methods approximate the posterior via the simulation of a Markov chain whose stationary distribution is the posterior distribution.
- Existing probabilistic programming systems cannot reason about the error introduced by simulating Markov chains for a finite number of steps — specially under composition of multiple approximate programs (or nested programs).

Contributions

1. Introduced a stat construct, that allows programmers to represent stationary distribution associated with a specified Markov chain, to the language proposed by Staton et al. 2016 and showed that the language constructs for conditioning and normalization are eliminable.
2. Under the assumptions of uniform ergodicity, gave quantitative error bounds for simulation based approximate implementation of stat.

Probabilistic programming language with the stationary construct

We extend the first order probabilistic language with probabilistic constructs sample, score, and norm.

First, we briefly review the semantics of probabilistic terms and introduce the semantics of the stat terms.

Sequencing, case and sampling terms:

\[
\text{[sample]}(t)_{\gamma,A} \equiv \{t\}_{\gamma,A}, \text{[let]} \ x = t_1 \ \text{in} \ t_2\}_{\gamma,A} \equiv \int_{[A]} [t_2]_{\gamma,x,A} [t_1]_{\gamma,dx} \text{[case]} a \text{of} \{(i,x) \Rightarrow t_i\}_{i\in I} \equiv [t]_{\gamma,\varepsilon} \text{ if } [a]_{\gamma} = (i,\varepsilon)
\]

Normalization and soft conditioning terms:

\[
\text{[score]}(a)_{\gamma,A} \equiv \begin{cases} |[a]_\gamma| & \text{if } A = \{\} \\ 0 & \text{otherwise} \end{cases}
\]

\[
\text{[norm]}(t)_{\gamma,A} \equiv \begin{cases} 0 & \text{if } [t]_{\gamma,A} \in (0,\infty) \\ \delta_{1,0}(\gamma)(A) & \text{otherwise} \end{cases}
\]

Stationary terms:

Given an initial distribution \(\pi_0\) and a probability transition kernel \(\lambda x.t\):

\[
\text{[stat]}(\pi_0, \lambda x.t)_{\gamma,A} = \begin{cases} \mu \{u: (0, u) \in A\} & \text{for } x \text{ a.e.} \ [t]_{\gamma}, \\ \delta_{1,0}(\gamma)(A) & \text{otherwise} \end{cases}
\]

Theorem (Soft conditioning and normalization terms are eliminable from the language)

Call the programming language defined before \(\mathcal{L}\). Let \(\mathcal{L}'\) be a programming language such that:

- the set of \(\mathcal{L}'\)-phrases is the full subset of \(\mathcal{L}\)-phrases that do not contain norm and score;
- the set of \(\mathcal{L}'\)-programs is the full subset of \(\mathcal{L}\)-programs that do not contain norm and score;
- the semantics of \(\mathcal{L}'\) is a restriction of \(\mathcal{L}\)'s semantics.

Then, every program that can be represented in \(\mathcal{L}\) can also be represented in \(\mathcal{L}'\).

Approximations in Probabilistic Programs:

a Compositional Analysis of Nonasymptotic MCMC

Ekansh Sharma, Daniel M. Roy

Approximate compilation of probabilistic programs

Problem (Failure of arbitrary approximate implementation.)

For some term \(\Gamma \vdash \text{stat}(t_0, \lambda x.t_1) : \mathcal{A} + 1\) if we know that \([t_1]_{\gamma,A}\) is an ergodic kernel that has a unique stationary distribution, it is possible to construct an approximate Markov transition kernel \(\lambda x.t'\) such that

\[
\exists \delta \in (0,1) \forall \gamma, x. \left\| [t_1]_{\gamma,A} - [t']_{\gamma,A} \right\|_{tv} \leq \delta.
\]

but

\[
\left\| [\text{stat}(t_0, \lambda x.t_1)]_\gamma - [\text{stat}(t_0, \lambda x.t')]_\gamma \right\|_{tv} = 1.
\]

Such an example is given in Proposition 1 of Roberts et al. (1998).

This tells us that stat construct is not continuous and we need to be careful how to approximate stat.

Now look at what goes on under the hood of a compiler:

Example compilation:

\[
\text{BetaPost}(N_F, N_P) := \text{norm} \left( \left| \text{score} \left( p^{N_F} (1 - p)^{N_P} \right) \right| \right) \left( \text{return} p \right)
\]

Compiling away the norm and score terms:

\[
\text{MHkern}(Q, p, N_F, N_P) := \text{let } p' = Q \text{ in }
\]\n
\[
\text{case sample(Bern(min) \left( \frac{1}{p'} \left( \frac{(1 - p')^{N_F} N_P}{p^N_F (1 - p')^{N_P}} \right) \right) \text{ of}}
\]\n
\[
\{ (0, T) \Rightarrow \text{return} p' \}
\]

\[
\{ (1, F) \Rightarrow \text{return} p \}
\]

\[
\text{BetaPost}(N_F, N_P) \sim \text{stat}(\text{Beta}(1,1), \lambda x. \text{MHkern}(\text{Beta}(1,1), p, N_F, N_P))
\]

Approximate implementation of the stat term by iterating:

\[
\text{ApproxBetaPost}(N_F, N_P, k) := \text{let } p = \text{Beta}(1,1) \text{ in }
\]\n
\[
\text{let } p = \text{MHkern}(\text{Beta}(1,1), p, N_F, N_P) \text{ in }
\]\n
\[
\text{let } p = \text{MHkern}(\text{Beta}(1,1), p, N_F, N_P) \text{ in }
\]\n
\[
\text{return} p \text{ in } (k \text{ steps})
\]

Theorem (Quantitative error bound for proposed approximations)

Let \(P\) be a probabilistic program in the language \(\mathcal{L}_{stat}\). Let \(\{\text{stat}(t_0, \lambda x.t_1)\}_{i \in I}\) be the set of all stationary terms in the program \(\bigcup_{i \in I} P : \mathcal{B}\) such that for all \(\gamma\), there exist constants \(\{C_i\}\) and \(\{\rho_i\}\) such that the Markov chain with the initial distribution \(\{t_0\}_\gamma\) and Markov transition kernel \(\{t_1\}_\gamma\) is uniformly ergodic with constants \(C_i\), \(\rho_i\).

Let \(P'\) be a program where for all \(i \in I\) and \(N_i \in \mathbb{N}\), \(\text{stat}(t_0, \lambda x.t_1)\) is replaced by \(\text{stat}(t_0, \lambda x.t_1)\). Then, there exist constants \(\{C_i\}_{i \in I}\) such that:

\[
\left\| \left[\text{stat}\right]_\gamma - [\text{stat}]_\gamma \right\|_{tv} \leq \sum_{i \in I} C_i \rho_i^{N_i}.
\]