Approximations in Probabilistic Programs: a Compositional Analysis of Nonasymptotic MCMC

Ekansh Sharma, Daniel M. Roy

VECTOR INSTITUTE — UNIVERSITY OF TORONTO

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:

$$[\![\mathbf{sample}(t)]\!]_{\gamma,A} \overset{\text{def}}{=} [\![t]\!]_{\gamma,A}\,, [\![\mathbf{let}\ x = t_1\ \mathbf{in}\ t_2]\!]_{\gamma,A} \overset{\text{def}}{=} \int_{[\![\mathbb{A}]\!]} [\![t_2]\!]_{\gamma,x,A}\,[\![t_1]\!]_{\gamma,dx}$$

$$\left[\!\!\left[\mathbf{case}\ a\ \mathbf{of}\ \{(i,x) \Rightarrow t_i\}_{i\in I}\right]\!\!\right] \overset{\text{def}}{=} [\![t_i]\!]_{\gamma,v} \quad \text{if } [\![a]\!]_\gamma = (i,v)$$

Normalization and soft conditioning terms:

$$[\![\mathbf{score}(a)]\!]_{\gamma,A} \overset{\text{def}}{=} \begin{cases} |[\![t]\!]_\gamma| & \text{if } A = \{()\} \\ 0 & \text{otherwise} \end{cases}$$

$$[\![\mathbf{norm}(t)]\!]_{\gamma,A} = \begin{cases} \frac{[\![t]\!]_{\gamma,\{u|(0,u)\in A\}}}{[\![t]\!]_{\gamma,[\![\mathbb{A}]\!]}} & \text{if } [\![t]\!]_{\gamma,[\![\mathbb{A}]\!]} \in (0,\infty) \\ \delta_{(1,())}(A) & \text{otherwise} \end{cases}$$

Stationary terms:

Given an initial distribution t_0 and a probability transition kernel $\lambda x.t_1$:

$$[\![\mathbf{stat}(t_0,\lambda x.t_1)]\!]_{\gamma,A} = \begin{cases} \mu(\{u : (0,u)\in A\}) & \text{if } \exists!\mu \in \mathcal{M}([\![\mathbb{A}]\!]) : \int_{[\![\mathbb{A}]\!]} [\![t_1]\!]^{(n)}(x,\cdot) \to_n \mu \\ & \text{for } x \text{ a.e. } [\![t_0]\!]_\gamma\,. \\ \delta_{(1,())}(A) & \text{otherwise} \end{cases}$$

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 $\mathcal{L}'$.

Approximate compilation of probabilistic programs

Problem (Failure of arbitrary approximate implementation.)

For some term $\Gamma \vdash_{p1} \mathbf{stat}(t_0,\lambda x.t_1) : \mathbb{A}+1$ if we know that $[\![t_1]\!]_{\gamma,x}$ is an ergodic kernel that has a unique stationary distribution, it is possible to construct an approximate Markov transition kernel $\lambda x.t_1'$ such that

$$\exists\delta\in(0,1)\forall\gamma,x.\left\|[\![t_1]\!]_{\gamma,x} - [\![t_1']\!]_{\gamma,x}\right\|_{\text{tv}} \le \delta,$$

but

$$\left\|[\![\mathbf{stat}(t_0,\lambda x.t_1)]\!]_\gamma - [\![\mathbf{stat}(t_0,\lambda x.t_1')]\!]_\gamma\right\|_{\text{tv}} = 1.$$

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

Example compilation:

$$\text{BetaPost}(N_F,N_P) := \mathbf{norm}\begin{pmatrix}\mathbf{let}\ p = \mathbf{Beta}(1,1)\ \mathbf{in} \\ \mathbf{score}\left(p^{N_F}(1-p)^{N_P}\right); \\ \mathbf{return}(p)\end{pmatrix}$$

Compiling away the **norm** and **score** terms:

$$\begin{aligned}\text{MHkern}(Q,p,N_F,N_P) :=&\ \mathbf{let}\ p' = Q\ \mathbf{in} \\ &\mathbf{case\ sample}\left(\text{Bern}\left(\min\left\{1, \frac{p'^{N_F}(1-p')^{N_P}}{p^{N_F}(1-p)^{N_P}}\right\}\right)\right)\ \mathbf{of} \\ &(0,T) \Rightarrow \mathbf{return}(p') \\ &\mid (1,F) \Rightarrow \mathbf{return}(p))\end{aligned}$$

$$\text{BetaPost}(N_F,N_P) \rightsquigarrow \mathbf{stat}(\mathbf{Beta}(1,1), \lambda p.\text{MHkern}(\mathbf{Beta}(1,1),p,N_F,N_P))$$

Approximate implementation of the **stat** term by iterating:

$$\begin{aligned}\text{ApproxBetaPost}(N_F,N_P,k) :=&\ \mathbf{let}\ p = \mathbf{Beta}(1,1)\ \mathbf{in} \\ &\left.\begin{aligned}&\mathbf{let}\ p = \text{MHkern}(\mathbf{Beta}(1,1),p,N_F,N_P)\ \mathbf{in} \\ &\mathbf{let}\ p = \text{MHkern}(\mathbf{Beta}(1,1),p,N_F,N_P)\ \mathbf{in} \\ &\vdots \\ &\mathbf{let}\ p = \text{MHkern}(\mathbf{Beta}(1,1),p,N_F,N_P); \mathbf{return}(p)\end{aligned}\right\}(k \text{ steps})\end{aligned}$$

Theorem (Quantitative error bound for proposed approximations)

Let P be a probabilistic program in the language $\mathcal{L}_{\mathbf{stat}}$. Let $\{\mathbf{stat}(t_{0i},\lambda x.t_{1i})\}_{i\in I}$ be the set of all stationary terms in the program $\varnothing \vdash_{p1} P : \mathbb{B}$ such that for all γ, there exist constants $\{C_i\}$ and $\{\rho_i\}$ such that the Markov chain with the initial distribution $[\![t_{0i}]\!]_\gamma$ and Markov transition kernel $[\![t_{1i})]\!]_\gamma$ is uniformly ergodic with constants C_i, ρ_i. Let P' be a program where for all $i\in I$ and $N_i\in\mathbb{N}$, $\mathbf{stat}(t_{0i},\lambda x.t_{1i})$ is replaced by $\phi(\mathbf{stat}(t_{0i},\lambda x.t_{1i}),N_i)$. Then, there exist constants $\{C_i'\}_{i\in I}$ such that

$$\left\|[\![P]\!]_\gamma - [\![P']\!]_\gamma\right\|_{\text{tv}} \le \sum_{i\in I} C_i'\rho_i^{N_i}.$$