A Lambda Calculus Foundation for Universal Probabilistic Programming

Johannes Borgström (Uppsala University)
Ugo Dal Lago (University of Bologna, INRIA)
Andrew D. Gordon (Microsoft Research, University of Edinburgh)
Marcin Szymczak (University of Edinburgh)

January 23, 2016

Introduction

We want to prove correct a variant of Metropolis-Hastings MCMC on program *traces* (sequences of random choices made during execution), in the line of the algorithm used by Church.

Why a formal correctness proof of Trace MCMC?

- . . . because there is none yet! (for a functional language)
- Can we really trust probabilistic languages and their inference engines?
- Machine learning used in safety-critical applications (medicine, autonomous vehicles etc.)
- Traces are highly nonstandard parameter spaces. Simple textbook proof for MH-MCMC does not apply.

Roadmap

To prove correctness of an inference algorithm for a probabilistic language we need:

- The syntax of the language
- A semantics of the language
- A rigorous definition of the algorithm
- A formal definition of "correct"

Roadmap

This paper consists of two parts:

- Semantics of a probabilistic lambda-calculus with continuous distribution, defined in two ways:
  - Distributional semantics- distribution on return values
  - Sampling-based semantics- distribution on random traces
- A formal proof of correctness of MH-MCMC on this language, with respect to the distributional semantics.
  - Still completing proofs of two measurability lemmas

Untyped lambda-calculus with continuous distributions

Let x, D, g range over countable sets of identifiers, distributions, primitive functions, respectively.

$$\begin{array}{rcl} V & ::= & c \mid x \mid \lambda x.M \\ M & ::= & V \mid M\ N \mid D(V_1, \ldots, V_{|D|}) \mid g(V_1, \ldots, V_{|g|}) \\ & & \quad \texttt{if}\ V\ \texttt{then}\ M\ \texttt{else}\ L \mid \texttt{fail} \\ G & ::= & V \mid \texttt{fail} \end{array}$$

We define a *metric space* on the space Λ of terms:

$$\begin{array}{rcll} d(c,d) & = & |c-d| & \\ d(x,x) & = & 0 & \\ d(\lambda x.M, \lambda x.N) & = & d(M,N) & \\ d(M\ N, L\ P) & = & d(M,L) + (N,P) & \ldots \end{array}$$

The metric space (Λ, d) gives rise to a topology and a Borel σ-algebra.

Example program

Using standard syntactic sugar for `let`.

$$
\begin{aligned}
&\texttt{let } p = \texttt{uniform}() \texttt{ in} \\
&\texttt{let } \mathit{flip} = \lambda\_.\texttt{uniform}() < p \texttt{ in} \\
&\texttt{if } (\mathit{flip}() = 0) \texttt{ and } (\mathit{flip}() = 1) \\
&\texttt{then } p \texttt{ else fail}
\end{aligned}
$$

Distributional Semantics- Small Step

Deterministic reduction: $M \to N$

$$E[(\lambda x.M)\ V] \xrightarrow{\text{det}} E[M\{V/x\}]$$

$$E[T] \xrightarrow{\text{det}} E[\texttt{fail}]$$

$$E[\texttt{fail}] \xrightarrow{\text{det}} \texttt{fail} \quad \text{if } E \text{ is not } [\cdot]$$

$$\dots$$

One-step evaluation: $M \to \mathscr{D}$

$$E[\mathrm{D}(\vec{c})] \to E\{\mu_{\mathrm{D}(\vec{c})}\}$$

$$E[M] \to \delta(E[N]) \text{ if } M \xrightarrow{\text{det}} N$$

Step-Indexed approximation semantics: $M \to^n \mathscr{D}$.

$$\frac{n > 0}{G \to_n \delta(G)}$$

$$\frac{}{M \to_0 \mathbf{0}}$$

$$\frac{M \to \mathscr{D} \qquad \{N \to_n \mathscr{E}_N\}_{N \in \mathrm{supp}(\mathscr{D})}}{M \to_{n+1} (A \mapsto \int \mathscr{E}_N(A)\ \mathscr{D}(dN))}$$

Semantics:

$$[\![M]\!]_{\Rightarrow} = \sup\{\mathscr{D} \mid M \to_n \mathscr{D}\}$$

Lemma

$\to$ is a subprobability kernel

Lemma

$\to^n$ is a subprobability kernel for every $n \geq 0$.

Distributional Semantics- Big Step

$$\frac{n>0}{G \Downarrow_n \delta(G)} \qquad \frac{}{M \Downarrow_0 \mathbf{0}} \qquad \frac{n>0}{T \Downarrow_n \delta(\texttt{error})} \qquad \frac{n>0}{\mathrm{D}(\vec{c}) \Downarrow_n \mu_{\mathrm{D}(\vec{c})}} \qquad \frac{n>0}{g(\vec{c}) \Downarrow_n \delta(\sigma_g(\vec{c}))}$$

$$\frac{M \Downarrow_n \mathscr{D}}{\texttt{if true then } M \texttt{ else } N \Downarrow_{n+1} \mathscr{D}} \qquad \frac{N \Downarrow_n \mathscr{D}}{\texttt{if false then } M \texttt{ else } N \Downarrow_{n+1} \mathscr{D}}$$

$$\frac{M \Downarrow_n \mathscr{D} \qquad N \Downarrow_n \mathscr{E} \qquad \{L\{V/x\} \Downarrow_n \mathscr{E}_{L,V}\}_{(\lambda x.L)\in\mathrm{supp}(\mathscr{D}),V\in\mathrm{supp}(\mathscr{E})}}{MN \Downarrow_{n+1} A \mapsto \mathscr{D}^{\mathcal{E}}(A) + \mathscr{D}(\mathbb{R})\cdot\delta(\texttt{error}) + \mathscr{D}(\mathcal{V}_\lambda)\cdot\mathscr{E}^{\mathcal{E}}(A) + \iint \mathscr{E}_{L,V}(A)\,\mathscr{D}^{\mathcal{V}_\lambda}(\lambda x.dL)\,\mathscr{E}^{\mathcal{V}}(dV)}$$

Semantics: $[\![M]\!]_{\Downarrow} = \sup\{\mathscr{D} \mid M \Downarrow_n \mathscr{D}\}$

Theorem

For every term M, $[\![M]\!]_{\Downarrow} = [\![M]\!]_{\Rightarrow}$.

Sampling Based Semantics - Pseudo-deterministic Evaluation

Small step: $(M, w, s) \to (M', w', s')$

$$\frac{M \xrightarrow{\text{det}} N}{(M, w, s) \to (N, w, s)} \qquad \frac{w' = \text{pdf}_{\text{D}}(\vec{c}, c) \qquad w' > 0}{(E[\text{D}(\vec{c})], w, s) \to (E[c], ww', s@[c])}$$

Big step: $M \Downarrow_w^s G$

$$\frac{G \in \mathcal{GV}}{G \Downarrow_1^{[]} G} \qquad \frac{w = \text{pdf}_{\text{D}}(\vec{c}, c) \qquad w > 0}{\text{D}(\vec{c}) \Downarrow_w^{[c]} c} \qquad \frac{}{g(\vec{c}) \Downarrow_1^{[]} \sigma_g(\vec{c})}$$

$$\frac{M \Downarrow_{w_1}^{s_1} \lambda x.P \qquad N \Downarrow_{w_2}^{s_2} V \qquad P[V/x] \Downarrow_{w_3}^{s_3} G}{M\ N \Downarrow_{w_1 w_2 w_3}^{s_1 @ s_2 @ s_3} G}$$

...

Proposition

$M \Downarrow_w^s G$ if and only if $(M, 1, []) \to^* (G, w, s)$.

Sampling Based Semantics: inspired by (Nori, Hur, Rajamani, Samuel 2013)

- Measurable space of program traces: $(\mathbb{S}, \mathcal{S})$, where:
  - $\mathbb{S} = \biguplus_{n \in \mathbb{N}} \mathbb{R}^n$
  - $\mathcal{S} = \{\biguplus_{n \in \mathbb{N}} H_n \mid H_n \in \mathcal{R}^n \quad \text{for all } n\}$
- Stock measure on program traces: $\mu(\biguplus_{n \in \mathbb{N}} H_n) = \sum_{n=1}^{\infty} \lambda_n(H_n)$
- Density function of a program M (w.r.t. stock measure on traces):

$$\mathbf{P}_M(s) = \begin{cases} w & \text{if } M \Downarrow_w^s G \text{ for some } G \\ 0 & \text{otherwise} \end{cases}$$

- Outcome of evaluation of M as a function of trace s:

$$\mathbf{O}_M(s) = \begin{cases} G & \text{if } M \Downarrow_w^s G \text{ for some } w \\ \texttt{fail} & \text{otherwise} \end{cases}$$

- A subprobability measure on program traces:

$$[\![M]\!]_{\Downarrow}^{\mathbb{S}}(A) = \int_A \mathbf{P}_M(s)\mu(ds)$$

- Can obtain measure on values by transformation: $[\![M]\!]_{\mathbb{S}} = [\![M]\!]_{\Downarrow}^{\mathbb{S}} \mathbf{O}_M^{-1}$

Theorem

$[\![M]\!]_{\mathbb{S}} = [\![M]\!]_{\Downarrow} = [\![M]\!]_{\Rightarrow}$

Recall: $[\![M]\!]_{\Rightarrow}$- Small-step distributional semantics
$[\![M]\!]_{\Downarrow}$- Big step distributional semantics

MCMC on General State Spaces (Green 1995, Tierney 1994, 1998)

Let (Ω, Σ) be an arbitrary measurable space. Suppose we want to sample from some distribution π on Σ.
Define a proposal kernel $Q(x, A) : \Omega \times \Sigma \to \mathbb{R}$ and a measurable acceptance function $\alpha(x, y) : \Omega \times \Omega \to [0, 1]$ such that the resulting Metropolis-Hastings transition kernel:

$$P(x, A) = \int_A \alpha(x, y) Q(x, dy) + \delta(x)(A) \int_\Omega (1 - \alpha(x, t)) Q(s, dt)$$

is *reversible* with respect to π:

$$\int_A P(x, B)\pi(dx) = \int_B P(y, A)\pi(dy)$$

for all $A, B \in \Sigma$.
Then π is the *stationary* distribution of the Markov chain with transition kernel P.
If $Q(x, A) = \int_A q(x, y)\mu(dy)$ and $\pi(A) = \int_A \dot{\pi}(x)\mu(dx)$, detailed balance equation simplifies.

MH-MCMC Inference

Idea: formalize the algorithm used by Church (or slightly simplified version thereof):

- Given trace $s = [s_1, \ldots, s_n]$ in program M, choose k s.t. $k \geq 0$, $k \leq n$ at random.
- Partially evaluate M under the trace $[s_1, \ldots, s_k]$, yielding M'.
- Evaluate M', sampling values $[t_{k+1}, \ldots, t_m]$ from target distributions on the way.
- Set $t = [s_1, \ldots, s_n t_{n+1}, \ldots, t_m]$, accept with probability $\alpha(s, t) = \min\{1, \frac{|t|}{|s|}\}$

Problem: the proposal kernel corresponding to this algorithm has no density!
Fixing a prefix would immediately set the integral to 0.
The lack of density makes the proof much harder. We have decided to leave it as further work and start with a kernel which has density.

MH-MCMC Inference- Take 2

Solution: update *all* elements of the trace, following the approach of (Hur, Nori et al, 2015).
Let $s = [s_1, \ldots, s_n]$ be the previous trace. For each i-th random choice:

- If $i < n$, draw $t_i = \texttt{Gaussian}(s_1, \sigma^2)$.
- Otherwise, draw t_i from target distribution.

Repeat until we get a generalized value and return trace t. Accept with probability

$$\alpha(s,t) = \begin{cases} 0 & \text{if } \mathbf{P}_M(t) = 0 \\ 1 & \text{if } \mathbf{P}_M(s)q(s,t) = 0 \\ \min\{1, \frac{\mathbf{P}_M(t)q(t,s)}{\mathbf{P}_M(s)q(s,t)}\} & \text{otherwise} \end{cases}$$

Inference- Take 2

This algorithm has the following transition kernel P:

$$\texttt{peval}(M,s) = \begin{cases} M & \text{if } s = [] \\ M' & \text{if } (M,1,[]) \Rightarrow (M_k, w_k, s_k) \to (M', w', s) \\ & \quad \text{for some } M_k, w_k, s_k, w' \text{ such that } s_k \neq s \\ \texttt{fail} & \text{otherwise} \end{cases}$$

$$q(s,t) = (\Pi_{i=1}^{k} \text{pdf}_{\text{Gaussian}}(s_i, \sigma^2, t_i)) \cdot \mathbf{P}_N(t_{k+1..|t|}) \text{ if } |t| \neq 0$$
$$\text{where } k = \min\{|s|,|t|\} \text{ and } N = \texttt{peval}(M, t_{1..k})$$

$$q(s,[]) = 1 - \int_A q(s,t)\,\mu(dt) \text{ where } A = \{t \mid |t| \neq 0\}$$

$$Q(s,A) = \int_A q(s,t)\,\mu(dt)$$

$$P(s,A) = \int_A \alpha(s,t)\,Q(s,dt) + \delta(s)(A) \cdot \int (1-\alpha(s,t))\,Q(s,dt)$$

Stationary distribution: $\pi(A) = [\![M]\!]^{\mathbb{S}}_{\Downarrow}(A)/[\![M]\!]^{\mathbb{S}}_{\Downarrow}(\mathbb{S})$ (normalized distribution on traces)

Definition of correctness

Define $P^n(x, A)$ to be the probability of reaching A from x in n steps:

$$\begin{aligned} P^0(s, A) &= \delta(s)(A) \\ P^{n+1}(s, A) &= \int P(t, A) P^n(s, dt) \end{aligned}$$

The *variational norm* is a measure of "closeness" of probability measures:

$$||\mu_1 - \mu_2|| = \sup_{A \in \Sigma} |\mu_1(A) - \mu_2(A)|$$

Let $T^n(s, A) = P^n(s, \mathbf{O}_M^{-1}(A))$ and $[\![M]\!]_{\mathcal{GV}}(A) = [\![M]\!](A)/[\![M]\!](\mathcal{GV})$.

The algorithm can be considered correct if for every trace s with $\mathbf{P}_M(s) \neq 0$,

$$\lim_{n \to \infty} ||T^n(s, \cdot) - [\![M]\!]_{\mathcal{GV}}|| = 0.$$

Proof of correctness

Theorem (Tierney 1994)

Let P be a Metropolis kernel (as given earlier). If π is the stationary distribution of P and P is π-irreducible and aperiodic, then

$$lim_{n\to\infty}||P^n(x,\cdot)-\pi|| = 0$$

Lemma (Strong Irreducibility, implies π-irreducibility)

If $\mathbf{P}_M(s) > 0$ and $[\![M]\!]^{\mathbb{S}}_{\Downarrow}(A) > 0$ then $P(s,A) > 0$.

Lemma (Aperiodicity)

P is π-aperiodic.

Then the above theorem from gives: $\lim_{n\to\infty} ||P^n(x,\cdot)-\pi|| = 0$

Theorem (Main Result)

For every trace s with $\mathbf{P}_M(s) \neq 0$, $\lim_{n\to\infty} ||T^n(s,\cdot) - [\![M]\!]_{\mathcal{GV}}|| = 0$.

Further work

- Finish proofs of two remaining technical lemmas (in second part)
- Translation of Church to the calculus
- Trial implementation
- Understanding conditioning
- Alternative inference algorithm, similar to Church
- Program MCMC in calculus itself