Continuous Probabilistic Programming[1]

Steven Holtzen
s.holtzen@northeastern.edu

October 17, 2023

[1] CS7470 Fall 2023: Foundations of Probabilistic Programming.

Some logistics:

- Friday Oct. 20 is Systems day: Please sign up to give a presentation on a system, and be sure to post your slides ahead of time in the Slack
- I will release and adjust the deadline for the continuous language this weekend based on what we've covered. It should be a pretty simple project.
- We will begin reading papers next week. See the instructions and consult the syllabus for which papers to read.

1 Cont: *A tiny continuous PPL*

Syntax:

```
v ::= real | bool
τ ::= 𝔹 | ℝ | Dist(ℝ)
e ::= unif | x ← e | x | v | e + e | e < e | e ∨ e | e ∧ e | ¬ e
```

- As usual, our recipe will have two parts: new denotation, new operational semantics. We begin with denotation.
- **Goal**: give a denotation for Cont that maps terms to probability distributions
- The `unif` keyword denotes a *uniform probability distribution on the interval* $[0, 1]$. This means we need to be able to handle probability distributions on the interval.
- For example, we can write the following program:

```
x ← unif;
return x < 1/2
```

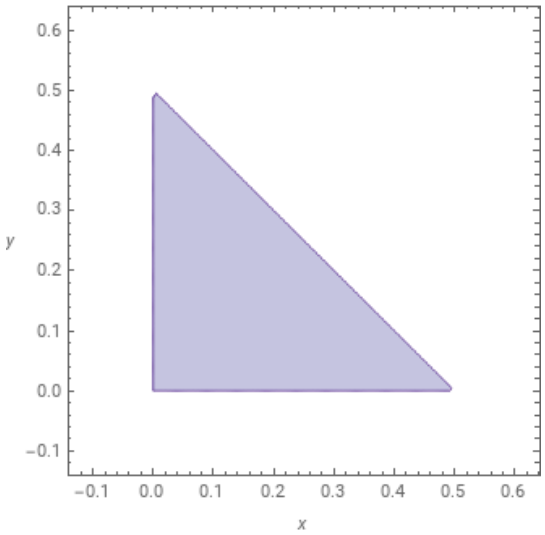
  How should we interpret this program? Intuitively, this program should return a distribution:

$$[\texttt{true} \mapsto 1/2, \texttt{false} \mapsto 1/2]$$

- What about this more interesting program:

```
x ← unif;
y ← unif;
return x + y < 1/2
```

This is significantly more challenging! The probability that this returns `true` is equal to the area of this triangle:



Put differently, it is an integral:

$$\int_0^1 \int_0^1 \mathbb{1}(x + y < 1/2)\, \mathrm{d}x\, \mathrm{d}y = 1/8 \tag{1}$$

Probability distributions on the interval

For an excellent discussion on the development of this section, see Axler [2020].

- **Goal**: Define a probability distribution on the sample space $\Omega = [0, 1]$
- Recall the challenge from last time: we cannot define a distribution naively on points because the probability of any particular real value $\omega \in \Omega$ should be 0.[2]
- Hence, we need to consider a subset of well-behaved semi-algebra of events on which to define a probability measure. A standard choice are the *collection of all open intervals* contained in $[0, 1]$:

$$\mathcal{J} = \{(l, u) \mid 0 \leq l < u \leq 1\} \cup \{\emptyset, \Omega\}$$

Why is this a good choice of events? Because we know a good way to define their probability:

$$\mu((l, u)) = u - l \qquad \mu(\emptyset) = 0 \qquad \mu(\Omega) = 1 \tag{2}$$

- The **Borel σ-algebra** is the smallest σ-algebra that contains $\mathcal{J}$. We denote the Borel σ-algebra as $\mathcal{B}$.[3]

[2] Why is this the case? Assume that there is a probability distribution $\mu : \Omega \to [0, 1]$ that assigns a non-zero probability to a countably-infinite set of points in $\omega \in \Omega$. Then, $\sum_{\omega \in \Omega} \mu(\omega) = \infty$, which is a contradiction.

[3] See Definition 2.27 of Axler [2020] for a discussion of why the notion of a smallest σ-algebra is well-founded.

- The unique probability measure $\lambda : \mathcal{B} \to [0,1]$ given by the extension of Eq. (2) to $\mathcal{B}$ is called the **Lebesgue measure** on the interval. For example:

$$\lambda\left((0,\frac{1}{2}) \cup (\frac{3}{4},1)\right) = 1/2 + 1/4 = 3/4.$$

- Now we will need a new generalized notion of random variable for this new setting. A random variable will be defined as a well-behaved function out of the sample space:

  Definition 1 (Measurable function)**.** *Let $(\Omega, \mathcal{F}, \mu)$ be a probability space, $(X, \mathcal{A})$ be a measurable space, and $f : \Omega \to X$ be a function. The function f is called* measurable *if, for any $A \in \mathcal{A}$, it is the case that $f^{-1}(A) \in \mathcal{F}$.*

  Measurable functions $f : [0,1] \to \mathbb{R}$ defined on $([0,1], \mathcal{B}, \lambda)$ are often called *Borel-measurable*. Most functions you have seen are Borel-measurable.[4]

[4] For instance, all continuous functions are Borel measurable.

- Measurable functions behave like random variables in that they induce probability measures on their co-domains via a *push-forward*:

  Definition 2. *Let $(\Omega, \mathcal{F}, \mu)$ be a probability space, $(X, \mathcal{A})$ be a measurable space, and $f : \Omega \to X$ be a measurable variable. Then, the* push-forward probability measure $\nu : \mathcal{A} \to [0,1]$ *is defined:*

$$\nu(A) = \mu(f^{-1}(A)). \tag{3}$$

2 *Denotation of* CONT

- These semantics are in the style of *Markov kernels* [Kozen, 1979, Barthe et al., 2020]
- The case for pure terms is relatively simple
- First we give the semantics:

The semantics of $x \leftarrow \mathsf{e}_1; \mathsf{e}_2$ is the bind operator for the Giry monad [Giry, 2006, Ramsey and Pfeffer, 2002]. This definition involves a *generalized integral* wrt. a base measure.

$$\begin{aligned}
[\![\mathtt{unif} : \mathtt{Dist}(\mathbb{R})]\!]\,(A) &= \lambda(A) \\
[\![\mathtt{return}\ \mathsf{e} : \mathtt{Dist}(\mathbb{R})]\!]\,(A) &= \mathbb{1}([\![\mathsf{e}]\!] \in A) = \delta_{[\![\mathsf{e}]\!]} \\
[\![x \leftarrow \mathsf{e}_1; \mathsf{e}_2]\!]\,(A) &= \mathbb{E}_{v \sim [\![\mathsf{e}_1]\!]}\Big[\,[\![\mathsf{e}_2[v/x]]\!]\,\Big] = \int_{\mathbb{R}} [\![\mathsf{e}_2[v/x]]\!]\,(A)\ \mathrm{d}\,[\![\mathsf{e}_1]\!]\,(v)
\end{aligned}$$

The *Diract-delta* $\delta_v(A)$ denotes a distribution that assigns a probability of 1 to any event that contains $v \in \Omega$.

- Integration wrt. the Lebesgue base measure is written $\int f(x)\ \mathrm{d}\lambda(x)$. It behaves just like Riemann integration (integrals from high school) if f is a suitably well-behaved function (i.e., continuous).
- Example:

$$\begin{aligned}\llbracket x \leftarrow \texttt{unif}; \texttt{return}\ x + x \rrbracket\,((0,1/2)) &= \int_0^1 \llbracket \texttt{return}\ v + v \rrbracket\,(v)((0,1/2))\ \mathrm{d}\lambda(v) \\ &= \int_0^1 \mathbb{1}(v + v \in (0,1/2))\ \mathrm{d}\lambda(v) \\ &= \int_0^{1/4} 1\ \mathrm{d}\lambda(v) \\ &= 1/4\end{aligned}$$

- Integration wrt. the Dirac measure δ_v is defined:

$$\int_{\mathbb{R}} f(x)\ \mathrm{d}\delta_v(x) \triangleq f(v). \tag{4}$$

  This lets us interpret programs with `return`:

$$\begin{aligned}\llbracket x \leftarrow \texttt{return}\ 1; \texttt{return}\ x + x \rrbracket\,(A) &= \int_{\mathbb{R}} \llbracket \texttt{return}\ v + v \rrbracket\,(A)\ \mathrm{d}\delta_1(v) \\ &= \int_{\mathbb{R}} \delta_{v+v}(A)\ \mathrm{d}\delta_1(v) \\ &= \delta_2(A).\end{aligned}$$

3 *Sampling semantics for* CONT

- These are very similar the sampling semantics for DISC from last time, except we need a different source of random bits
- We will entropy $\sigma \in [0,1]^{\mathbb{N}}$, i.e. σ is a countably infinite stream of real values drawn from the interval. We again define π_L and π_R as ways of splitting the entropy into independent sets.
- Then we will define a big-step relation $\sigma \vdash \mathsf{e} \Downarrow v$:

$$v :: \sigma \vdash \texttt{unif} \Downarrow v \qquad \frac{\pi_L(\sigma) \vdash \mathsf{e}_1 \Downarrow v_1 \qquad \pi_R(\sigma) \vdash \mathsf{e}_2[v_1/x] \Downarrow v_2}{\sigma \vdash x \leftarrow \mathsf{e}_1; \mathsf{e}_2 \Downarrow v_2}$$

$$\sigma \vdash \texttt{return}\ \mathsf{v} \Downarrow v$$

Theorem 1. *Assume* e *is a well-typed closed term and* Pr *is the distribution on the entropy space. Then, for any event* $A \in \mathcal{B}$*, we have that:*

$$\mathbb{E}_{\sigma \sim \Pr}\Big[\mathbb{1}[\mathit{eval}(\mathsf{e}, \sigma) \in A]\Big] = \llbracket \mathsf{e} \rrbracket\,(A). \tag{5}$$

4 *Continuous observation*

Add a new `observe` keyword to CONT:

```
v ::= real | bool
τ ::= 𝔹 | ℝ | Dist(ℝ)
e ::= unif | x ← e | x | v | e + e | e ∗ e | e < e | e ∨ e | e ∧ e | ¬ e
   | observe e; e
```

This keyword is surprisingly powerful. Suppose we want to infer something about the distribution of people's heights. Suppose we assume that people are between 65 and 73 inches tall, and that people vary from the true value by 1 inch. Then, we observe that someone's height is between 71 and 73 inches, and want to update prior beliefs about how people's heights are distributed. We can model this as the following program:

```
h ← unif;
height_dist ← (h ∗ 4) + 69
eps1 ← unif;
observe 71 < height_dist + eps1 < 73;
return height_dist
```

References

Sheldon Axler. *Measure, integration & real analysis*. Springer Nature, 2020.

Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva. *Foundations of probabilistic programming*. Cambridge University Press, 2020.

Michele Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis: Proceedings of an International Conference Held at Carleton University, Ottawa, August 11–15, 1981*, pages 68–85. Springer, 2006.

Dexter Kozen. Semantics of probabilistic programs. In *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, pages 101–114. IEEE, 1979.

Norman Ramsey and Avi Pfeffer. Stochastic lambda calculus and monads of probability distributions. In *Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 154–165, 2002.