Language of Expressions Involving Random Variables

November 14, 2000

1 Random Variables

First we must consider what a random variable is. A statistics textbook describes them as functions $X : Events \to \mathcal{R}$. To generalize this let's say that $X : Events \to D$ for some domain D. Now we are not just interested in the value $X(e)$ for a particular event, we would like to know the probability $\Pr(X(e) = x_0)$ or the distribution of $X(e)$ over $Events$. This distribution can be characterized by a PDF $f_X : D \to \mathcal{R}^+$. The usual conditions on PDF's apply. Events over discrete domains will require a different treatment.

Definitions:

$$
\begin{aligned}
RandV &= \{(X, f_X) \mid X : Events \to D \wedge f_X : D \to \mathcal{R}^+\} && (1)\\
[[-]] &: Exp \to RandV && (2)
\end{aligned}
$$

2 Syntax of Expressions

$$
\begin{aligned}
ExpId &\rightarrow Id \mid Exp && (3)\\
Exp &\rightarrow RandV && (4)\\
&\quad \mid Exp\ BinOp\ Exp && (5)\\
&\quad \mid \textbf{probloop}\ (ExpList)\ Body && (6)\\
Body &\rightarrow \lambda Id, \ldots, Id\ .\ ExpId && (7)\\
ExpList &\rightarrow \epsilon \mid Exp \mid Exp, ExpList && (8)\\
BinOp &\rightarrow + \mid * && (9)\\
& && (10)
\end{aligned}
$$

3 Semantics of Expressions

$$
\begin{aligned}
[[E_1 * E_2]] &= (Z, f_Z) && (11)\\
&\quad \text{where}\\
&\quad (X, f_X) = [[E_1]]\\
&\quad (Y, f_Y) = [[E_2]]\\
&\quad Z = \lambda e.(X(e)(\cdot_D)Y(e))\\
&\quad f_Z = \lambda z.(\frac{d}{dz}(\iint_R f_X(x) \cdot f_Y(y) dx dy))\\
&\quad R = \{(x, y) \in D^2 \mid x(\cdot_D)y \leq_D z\}\\
[[E_1 + E_2]] &= (Z, f_Z) && (12)\\
&\quad \text{where}\\
&\quad (X, f_X) = [[E_1]]\\
&\quad (Y, f_Y) = [[E_2]]\\
&\quad Z = \lambda e.(X(e)(+_D)Y(e))\\
&\quad f_Z = \lambda z.(\frac{d}{dz}(\iint_R f_X(x) \cdot f_Y(y) dx dy))\\
&\quad R = \{(x, y) \in D^2 \mid x(+_D)y \leq_D z\}
\end{aligned}
$$

$$
\begin{aligned}
[[E_1 \# E_2]] &= (Z, f_Z) \\
& \text{where} \\
& (X, f_X) = [[E_1]] \\
& (Y, f_Y) = [[E_2]] \\
& Z = X?? \\
& f_Z = \lambda z.(\frac{f_X(z) \cdot f_Y(z)}{\int_{-\infty}^{+\infty} f_X(z) \cdot f_Y(z) dz})
\end{aligned}
\tag{13}
$$

$$
\begin{aligned}
[[\mathit{probloop}\ (E_1, \ldots, E_k)\ \mathit{body}]] &= (Z, f_Z) \\
\text{where} & \\
(Y_i, f_{Y_i}) &= [[E_i]] \\
Z &= \lambda e.([[\mathit{body}]]\,(Y_1(e), \ldots, Y_k(e)) \\
f_Z &= \lambda z.(\frac{d}{dz}(\int \cdots \int_R f_{Y_1}(y_1), \ldots, f_{Y_k}(y_k) dy_1, \ldots, dy_k)) \\
R &= \{(y_1, \ldots, y_k) \in D^k \mid [[\mathit{body}]]\,(y_1, \ldots, y_k) \leq_D z\}
\end{aligned}
\tag{14}
$$

The assumption of independence manifests itself as a substitution of $f_X \cdot f_Y$ for $f_{X,Y}$ in the equations for $+, *$. Thus independence of random variables is easy to express in the semantic definitions. Instead, it turns out that dependence is the problem.

The definition of probloop uses a similar substitution because it assumes that the $E_1, \ldots, E_k$ are initially independent. For "simple" body functions there is nowhere for dependency to originate. For example the following are equivalent:

$$
[[\mathit{probloop}(x, y)\lambda(x_0, y_0).(x_0(\cdot_D)y_0)]] = [[x * y]] \tag{15}
$$

Dependencies arise from complexity of the $[[\mathit{body}]]$ function. Consider the equation for the region R in the definition of probloop.

$$
R = \{(y_1, \ldots, y_k) \in D^k \mid [[\mathit{body}]]\,(y_1, \ldots, y_k) \leq_D z\}
$$

In general, R could be very complicated, with numerous "holes" or other irregularities that make analytical integration impossible. It is easy to verify whether $(y_1, \dots, y_k) \in R$, but very hard to generate the set or decompose it along the dimensions $(dy_1, \dots, dy_k)$. This is where a sample-based representation comes in handy.