

EQUATIONAL REASONING FOR PROBABILISTIC PROGRAMMING

CHUNG-CHIEH SHAN, INDIANA UNIVERSITY

1. THE FIELDS

	Programming	Probability
Theoretical	Induction	Integral
Practical	Interpreter	Inference

2. THE TASKS

Whenever we're unsure about something, represent our uncertain knowledge as a distribution.

2.1. The table game. (Eddy 2004)

```
casino : M Bool
casino  $\triangleq$  do {p  $\leftarrow$  uniform 0 1;
             a1  $\leftarrow$  binomial 8 p;
             ()  $\leftarrow$  guard (a1 = 5);
             a2  $\leftarrow$  binomial 3 p;
             return (a2  $\geq$  1)}
```

2.2. Inferring behavior from text-message data. (Davidson-Pilon 2016)

```
texting : M ( $\overline{\mathbb{R}}_+ \times \overline{\mathbb{R}}_+$ )
texting  $\triangleq$  do {r1  $\leftarrow$  exponential 37;
             r2  $\leftarrow$  exponential 42;
             t0  $\leftarrow$  counting 1 70;
              $\vec{c}$   $\leftarrow$  mapM ( $\lambda t$ . poisson (if t < t0 then r1 else r2))
                    [1..70];
             ()  $\leftarrow$  guard ( $\vec{c}$  = [13, 24, ..]);
             return (r1, r2)}
```

2.3. Observing a noisy draw from a normal distribution.

```
helloWrong :  $\mathbb{R} \rightarrow M \mathbb{R}$ 
helloWrong y0  $\triangleq$  do {x  $\leftarrow$  normal 0 1;
                    y  $\leftarrow$  normal x 1;
                    ()  $\leftarrow$  guard (y = y0); -- WRONG!
                    z  $\leftarrow$  normal x 1;
                    return z}

helloRight :  $\mathbb{R} \rightarrow M \mathbb{R}$ 
helloRight y0  $\triangleq$  do {x  $\leftarrow$  normal 0 1;
                    ()  $\leftarrow$  factor  $\frac{e^{-(y_0-x)^2/2}}{\sqrt{2 \cdot \pi}}$ ;
                    z  $\leftarrow$  normal x 1;
                    return z}

helloJoint : M ( $\mathbb{R} \times \mathbb{R}$ )
helloJoint  $\triangleq$  do {x  $\leftarrow$  normal 0 1;
                 y  $\leftarrow$  normal x 1;
                 z  $\leftarrow$  normal x 1;
                 return (y, z)} -- Ready to disintegrate
```

3. THE EQUATIONS

3.1. Nondeterminism and weights.

$$\text{binomial } 2 p = \begin{aligned} & (p \quad \odot p \quad \odot \text{return } 2) \oplus \\ & (p \quad \odot (1-p) \odot \text{return } 1) \oplus \\ & ((1-p) \odot p \quad \odot \text{return } 1) \oplus \\ & ((1-p) \odot (1-p) \odot \text{return } 0) \end{aligned} \quad (1)$$

$$= \begin{aligned} & (p^2 \quad \odot \text{return } 2) \oplus \\ & (2p(1-p) \odot \text{return } 1) \oplus \\ & ((1-p)^2 \odot \text{return } 0) \end{aligned} \quad (2)$$

3.2. From rejection sampling to importance sampling. (MacKay 1998)

$$\text{casino} = \text{do } \{p \leftarrow \text{uniform } 0 \ 1; \quad (3)$$

$$a_1 \leftarrow \text{binomial } 8 \ p;$$

$$() \leftarrow \text{guard } (a_1 = 5);$$

$$((1 - (1 - p)^3) \odot \text{return True}) \oplus$$

$$((1 - p)^3 \odot \text{return False})\}$$

$$= \text{do } \{p \leftarrow \text{uniform } 0 \ 1; \quad (4)$$

$$() \leftarrow \text{factor } (56 \cdot p^5 \cdot (1 - p)^3);$$

$$((1 - (1 - p)^3) \odot \text{return True}) \oplus$$

$$((1 - p)^3 \odot \text{return False})\}$$

In general, if $m = r \odot n$, then we say that the function r is a *density* or *Radon-Nikodym derivative* of m with respect to n . If we know how to sample n , then r tells us how to importance-sample m using the proposal distribution n .

3.3. Density facts. If r is a density of m with respect to n , then $r \circ f^{-1}$ is a density of $\text{fmap } f \ m$ with respect to $\text{fmap } f \ n$ whenever f is invertible.

If r is a density of m with respect to n , then $\text{recip} \circ r$ is a density of n with respect to m . Here recip is the reciprocal function

$\lambda x. 1/x$, and a side condition is that the reciprocal must be defined almost everywhere:

$$\text{do } \{x \leftarrow n; \text{guard } \neg(0 < r \ x < \infty)\} = \text{fail} \quad (5)$$

3.4. Conjugate prior and density recognition.

$$\text{casino} = (1/9) \odot \text{do } \{p \leftarrow \text{beta } 6 \ 4; \quad (6)$$

$$((1 - (1 - p)^3) \odot \text{return True}) \oplus$$

$$((1 - p)^3 \odot \text{return False})\}$$

$$\text{helloRight } y_0 = \frac{e^{-y_0^2/4}}{\sqrt{4 \cdot \pi}} \odot \text{do } \{x \leftarrow \text{normal } (y_0/2) \ (1/\sqrt{2}); \quad (7)$$

$$z \leftarrow \text{normal } x \ 1;$$

$$\text{return } z\}$$

3.5. Variable elimination and integration.

$$\text{casino} = (1/9) \odot (((10/11) \odot \text{return True}) \oplus \quad (8)$$

$$((1/11) \odot \text{return False}))$$

$$\text{helloRight } y_0 = \frac{e^{-y_0^2/4}}{\sqrt{4 \cdot \pi}} \odot \text{normal } (y_0/2) \ \sqrt{3/2} \quad (9)$$

3.6. From density to disintegration. (Shan and Ramsey 2017)

$$\text{helloJoint} = \text{lebesgue } (-\infty) \ \infty \ \otimes \ \text{helloRight} \quad (10)$$

Generalize helloRight to a *Kalman filter*, such as a function

$$f : \text{State} \rightarrow \mathbb{R} \rightarrow \text{State}$$

(where $\text{State} = \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+$) satisfying

$$\text{interpret } (f \ w \ c \ d) = \text{do } \{x \leftarrow \text{interpret } w \ c \ d; \quad (11)$$

$$\frac{e^{-(y_0-x)^2/2}}{\sqrt{2 \cdot \pi}} \odot \text{normal } x \ 1\}$$

(where $\text{interpret } (w, c, d) = w \odot \text{normal } c \ d$).

3.7. Markov chain Monte Carlo. (MacKay 1998; Tierney 1998; McElreath 2017)

Suppose we want samples from a given *target distribution*

$$p : \mathbb{M} \alpha$$

but it is inefficient or weighted as a sampler. To use Markov chain Monte Carlo, we seek a *transition kernel*

$$k : \alpha \rightarrow \mathbb{M} \alpha,$$

and iterate it to perform a random walk in the state space α . Our k should return a probability measure that is efficient and unweighted as a sampler. Moreover, it should satisfy *detailed balance*:

$$p \otimes k = k \otimes p \quad (12)$$

(Intuition: if $p \otimes k = k \otimes p$ then $p \ggg k = p$.)

Metropolis-Hastings is a way to construct k from a *proposal distribution*

$$q : \alpha \rightarrow \mathbb{M} \alpha.$$

Like the k we want, the q we provide should return a probability measure that is efficient and unweighted as a sampler, but q need not satisfy detailed balance (in other words, it is fine if $p \otimes q \neq q \otimes p$).

To use Metropolis-Hastings given q , first find a density r of $p \otimes q$ with respect to $q \otimes p$. That is, find

$$r : (\alpha \times \alpha) \rightarrow \overline{\mathbb{R}}_+$$

such that

$$p \otimes q = r \odot (q \otimes p). \quad (13)$$

Then let

$$\alpha(x, y) = \min\{1, r(x, y)\} \quad (14)$$

$$\begin{aligned} k \text{ old} = & \text{do } \{ \text{new} \sim q \text{ old}; \\ & \text{accept} \leftarrow \text{bernoulli}(\alpha(\text{new}, \text{old})); \\ & \text{return (if accept then new else old)} \} \end{aligned} \quad (15)$$

4. THE INTERPRETATIONS: WHAT ARE WE EQUATING?

4.1. Denotational semantics. (Culpepper and Cobb 2017; Staton 2017; Heunen et al. 2017; Ścibior et al. 2018)

Good for declaring what we want to compute. Equality is a congruence.

Measures are equivalent to integrators. Easy to understand as continuation-passing style.

4.2. Operational semantics: samplers. Good for implementing algorithms. But what kind of samplers?

4.2.1. Randomized samplers (“Monte Carlo methods”) vs deterministic code.

- Randomized samplers, such as Eddy’s (2004) samplers.
- Deterministic code, such as Eddy’s (2004) exact formula.

4.2.2. Weighted vs unweighted results.

- Weighted results, as from importance sampling, are superposed over time.

Example use: histograms and other forms of expectation estimation.

- Unweighted results, as from rejection sampling, are used committally.

Example use: deciding how to drive or where to visit next in a graph.

- Converting unweighted result stream to weighted is trivial.
- Converting weighted result stream to unweighted requires bound on weight.

4.2.3. Efficient vs inefficient algorithms.

- Sure we want samples fast, but slower samples can be more accurate.

5. THE LANGUAGE

return	$: \alpha \rightarrow \mathbb{M} \alpha$		
$\gg=$ ($\gg=$)	$: \mathbb{M} \alpha \rightarrow (\alpha \rightarrow \mathbb{M} \beta) \rightarrow \mathbb{M} \beta$	$m \gg= k$	$= \text{do } \{x \leftarrow m; y \leftarrow k \ x; \text{return } y\}$
\gg (\gg)	$: \mathbb{M} \alpha \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} \beta$	$m \gg n$	$\triangleq \text{do } \{ _ \leftarrow m; y \leftarrow n; \text{return } y\}$
\Rightarrow (\Rightarrow)	$: (\beta \rightarrow \mathbb{M} \alpha) \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} \beta$	$k \Rightarrow n$	$\triangleq \text{do } \{y \leftarrow n; _ \leftarrow k \ y; \text{return } y\}$
$\otimes=$ ($\otimes=$)	$: \mathbb{M} \alpha \rightarrow (\alpha \rightarrow \mathbb{M} \beta) \rightarrow \mathbb{M} (\alpha \times \beta)$	$m \otimes= k$	$\triangleq \text{do } \{x \leftarrow m; y \leftarrow k \ x; \text{return } (x, y)\}$
\otimes (\otimes)	$: \mathbb{M} \alpha \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} (\alpha \times \beta)$	$m \otimes n$	$\triangleq \text{do } \{x \leftarrow m; y \leftarrow n; \text{return } (x, y)\} = \text{do } \{y \leftarrow n; x \leftarrow m; \text{return } (x, y)\}$
$\Rightarrow \otimes$ ($\Rightarrow \otimes$)	$: (\beta \rightarrow \mathbb{M} \alpha) \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} (\alpha \times \beta)$	$k \Rightarrow \otimes n$	$\triangleq \text{do } \{y \leftarrow n; x \leftarrow k \ y; \text{return } (x, y)\}$
fmap	$: (\alpha \rightarrow \beta) \rightarrow \mathbb{M} \alpha \rightarrow \mathbb{M} \beta$	fmap $f \ m$	$\triangleq m \gg= (\text{return } \circ f)$
\oplus ($\langle + \rangle$)	$: \mathbb{M} \alpha \rightarrow \mathbb{M} \alpha \rightarrow \mathbb{M} \alpha$		
fail	$: \mathbb{M} \alpha$		
guard	$: \text{Bool} \rightarrow \mathbb{M} \text{Unit}$	guard b	$\triangleq \text{if } b \text{ then return } () \text{ else fail}$
factor	$: \overline{\mathbb{R}}_+ \rightarrow \mathbb{M} \text{Unit}$		
\odot ($*\rangle$)	$: \overline{\mathbb{R}}_+ \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} \beta$	$p \odot n$	$\triangleq \text{factor } p \gg n$
$\Rightarrow \odot$ ($\Rightarrow * \rangle$)	$: (\beta \rightarrow \overline{\mathbb{R}}_+) \rightarrow \mathbb{M} \beta \rightarrow \mathbb{M} \beta$	$r \Rightarrow \odot n$	$\triangleq (\text{factor } \circ r) \Rightarrow n$
bernoulli	$: [0, 1] \rightarrow \mathbb{M} \mathbb{N}$	bernoulli p	$= (p \odot \text{return } 1) \oplus ((1 - p) \odot \text{return } 0)$
binomial	$: \mathbb{N} \rightarrow [0, 1] \rightarrow \mathbb{M} \mathbb{N}$	binomial $i \ p$	$= \text{fmap sum } (\text{replicateM } i \ (\text{bernoulli } p))$
counting	$: \overline{\mathbb{N}} \rightarrow \overline{\mathbb{N}} \rightarrow \mathbb{M} \mathbb{N}$		
geometric	$: [0, 1] \rightarrow \mathbb{M} \mathbb{N}$	geometric p	$= (\lambda i. (1 - p) \cdot p^i) \Rightarrow \odot \text{counting } 0 \ \infty$
poisson	$: \mathbb{R}_+ \rightarrow \mathbb{M} \mathbb{N}$	poisson r	$: (\lambda i. r^i / e^r / i!) \Rightarrow \odot \text{counting } 0 \ \infty$
lebesgue	$: \overline{\mathbb{R}} \rightarrow \overline{\mathbb{R}} \rightarrow \mathbb{M} \mathbb{R}$		
uniform	$: \mathbb{R} \rightarrow \mathbb{R} \rightarrow \mathbb{M} \mathbb{R}$	uniform $x \ y$	$= (1 / (y - x)) \odot \text{lebesgue } x \ y$
beta	$: \mathbb{R}_+ \rightarrow \mathbb{R}_+ \rightarrow \mathbb{M} [0, 1]$	beta $a \ b$	$= (\lambda p. p^{a-1} \cdot (1 - p)^{b-1} / \text{B}(a, b)) \Rightarrow \odot \text{lebesgue } 0 \ 1$
exponential	$: \mathbb{R}_+ \rightarrow \mathbb{M} \mathbb{R}_+$	exponential l	$= \text{do } \{x \leftarrow \text{lebesgue } 0 \ \infty; e^{-x} \odot \text{return } (l \cdot x)\}$
normal	$: \mathbb{R} \rightarrow \mathbb{R}_+ \rightarrow \mathbb{M} \mathbb{R}$	normal $c \ d$	$= \text{do } \{x \leftarrow \text{lebesgue } (-\infty) \ \infty; (e^{-x^2/2} / \sqrt{2 \cdot \pi}) \odot \text{return } (c + d \cdot x)\}$

6. THE LAWS

6.1. $\gg=$ and return form a commutative monad.

$$\text{return } x \gg= k = k \ x \quad (16)$$

$$m \gg= \text{return} = m \quad (17)$$

$$(m \gg= k) \gg= l = m \gg= \lambda x. (k \ x \gg= l) \quad (18)$$

$$\text{do } \{x \leftarrow m; y \leftarrow n; k \ x \ y\} = \text{do } \{y \leftarrow n; x \leftarrow m; k \ x \ y\} \quad (19)$$

(Generalize $\otimes=$ to countable products?)

6.2. \oplus and fail form a commutative monoid.

$$\text{fail} \oplus m = m \quad (20)$$

$$= m \oplus \text{fail} \quad (21)$$

$$(m \oplus n) \oplus o = m \oplus (n \oplus o) \quad (22)$$

$$m \oplus n = n \oplus m \quad (23)$$

(Generalize \oplus to countable sums?)

6.5. Conjugate priors. (derived by algebra)

$$(\lambda p. p^{a'} \cdot (1-p)^{b'}) \circledast \text{beta } a \ b = \frac{\text{B}(a+a', b+b')}{\text{B}(a, b)} \circledast \text{beta } (a+a') \ (b+b') \quad (32)$$

$$\left(\lambda x. \frac{e^{-(x-c')^2/d'^2/2}}{\sqrt{2 \cdot \pi \cdot d'}} \right) \circledast \text{normal } c \ d = \frac{e^{-(c-c')^2/(d^2+d'^2)/2}}{\sqrt{2 \cdot \pi \cdot (d^2+d'^2)}} \circledast \text{normal } \frac{c \cdot d^{-2} + c' \cdot d'^{-2}}{d^{-2} + d'^{-2}} \frac{1}{\sqrt{d^{-2} + d'^{-2}}} \quad (33)$$

6.6. Probability measures. (derived by integral calculus)

$$\text{bernoulli } p \gg n = n \quad (34)$$

$$\text{geometric } p \gg n = n \quad (35)$$

$$\text{poisson } r \gg n = n \quad (36)$$

$$\text{uniform } x \ y \gg n = n \quad (37)$$

$$\text{beta } a \ b \gg n = n \quad (38)$$

$$\text{exponential } l \gg n = n \quad (39)$$

$$\text{normal } c \ d \gg n = n \quad (40)$$

6.3. \oplus and fail distribute over $\gg=$.

$$(m \oplus n) \gg= k = (m \gg= k) \oplus (n \gg= k) \quad (24)$$

$$m \gg= \lambda x. (k \ x \oplus l \ x) = (m \gg= k) \oplus (m \gg= l) \quad (25)$$

$$\text{fail} \gg= k = \text{fail} \quad (26)$$

$$m \gg \text{fail} = \text{fail} \quad (27)$$

6.4. factor is an isomorphism between $\overline{\mathbb{R}}_+$ and \mathbb{M} Unit.

$$\text{factor } p \gg \text{factor } q = \text{factor } (p \cdot q) \quad (28)$$

$$\text{factor } p \oplus \text{factor } q = \text{factor } (p + q) \quad (29)$$

$$\text{return } () = \text{factor } 1 \quad (30)$$

$$\text{fail} = \text{factor } 0 \quad (31)$$

(Treat $\overline{\mathbb{R}}_+$ as synonym for \mathbb{M} Unit?)

6.7. Change of variables. (derived by integral calculus)

$$\text{fmap } (\lambda x. -\log x) \ (\text{uniform } 0 \ 1) = \text{exponential } 1 \quad (41)$$

$$\begin{aligned} \text{fmap } (\lambda x. c + d \cdot x) \ (\text{lebesgue } a \ b) \\ = (1/d) \circledast \text{lebesgue } (c + d \cdot a) \ (c + d \cdot b) \end{aligned} \quad (42)$$

REFERENCES

- Culpepper, Ryan, and Andrew Cobb. 2017. Contextual equivalence for probabilistic programs with continuous random variables and scoring. In *Programming languages and systems: Proceedings of ESOP 2017, 26th European symposium on programming*, ed. Yang Hongseok, 368–392. Lecture Notes in Computer Science 10201, Berlin: Springer.
- Davidson-Pilon, Cameron. 2016. *Bayesian methods for hackers: Probabilistic programming and Bayesian inference*. Boston: Addison-Wesley.
- Eddy, Sean R. 2004. What is Bayesian statistics? *Nature Biotechnology* 22(9):1177–1178.
- Heunen, Chris, Ohad Kammar, Sam Staton, and Hongseok Yang. 2017. A convenient category for higher-order probability theory. In *LICS 2017: Proceedings of the 32nd symposium on logic in computer science*, 1–12. Washington, DC: IEEE Computer Society Press.
- MacKay, David J. C. 1998. Introduction to Monte Carlo methods. In *Learning and inference in graphical models*, ed. Michael I. Jordan. Dordrecht: Kluwer. Paperback: *Learning in Graphical Models*, MIT Press.
- McElreath, Richard. 2017. Markov chains: Why walk when you can flow? <http://eleanth.org/blog/2017/11/28/build-a-better-markov-chain/>.
- Ścibior, Adam, Ohad Kammar, Matthijs Vákár, Sam Staton, Hongseok Yang, Yufei Cai, Klaus Ostermann, Sean K. Moss, Chris Heunen, and Zoubin Ghahramani. 2018. Denotational validation of higher-order Bayesian inference. In *POPL '18: Conference record of the annual ACM symposium on principles of programming languages*. New York: ACM Press.
- Shan, Chung-chieh, and Norman Ramsey. 2017. Exact Bayesian inference by symbolic disintegration. In *POPL '17: Conference record of the annual ACM symposium on principles of programming languages*, 130–144. New York: ACM Press.
- Staton, Sam. 2017. Commutative semantics for probabilistic programming. In *Programming languages and systems: Proceedings of ESOP 2017, 26th European symposium on programming*, ed. Yang Hongseok, 855–879. Lecture Notes in Computer Science 10201, Berlin: Springer.
- Tierney, Luke. 1998. A note on Metropolis-Hastings kernels for general state spaces. *The Annals of Applied Probability* 8(1): 1–9.